

## IS200x/IS201x/IS202x UART Command SOP



ISSC Technologies Corp.

### Revision History:

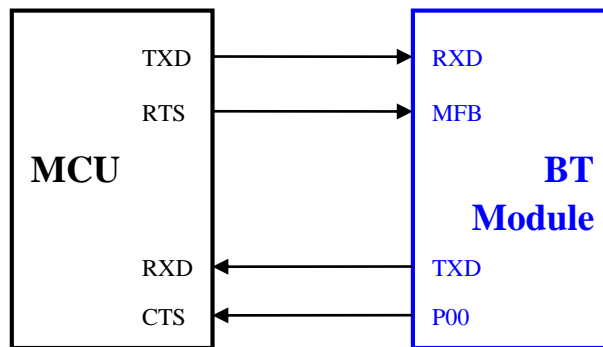
Date	Revision Content	Revised By	Version
2014/8/25	First edition	HP	1.0
2014/8/29	Revise profile link back command length: 03 -> 02	HP	1.1
2014/9/2	Revise power on procedure	HP	1.2
2014/10/24	Modify power on UI configuration	HP	1.3
2014/10/31	Remove NFC action(disconnect)	HP	1.4

**Description:**

This document is an operating procedure for customer to implement their **MCU** codes which accomplish communications with ISSC 200x, 201x, and 202x related SPK modules (BM13/BM15/BM20/BM23/BM25). The communication interface is UART plus CTS and RTS GPIO signals.

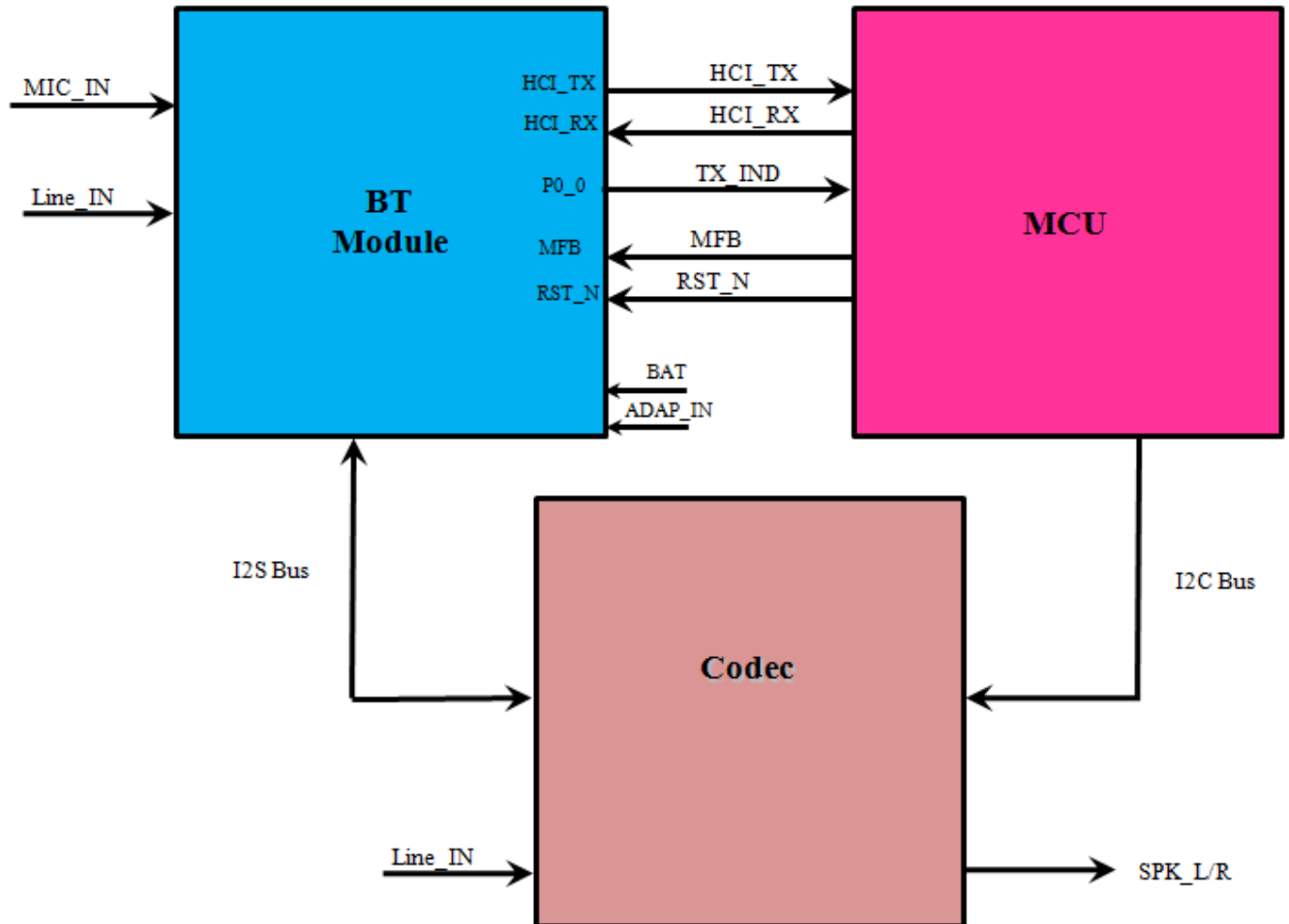
*Pin Definition:*

The pin definitions are described as below.



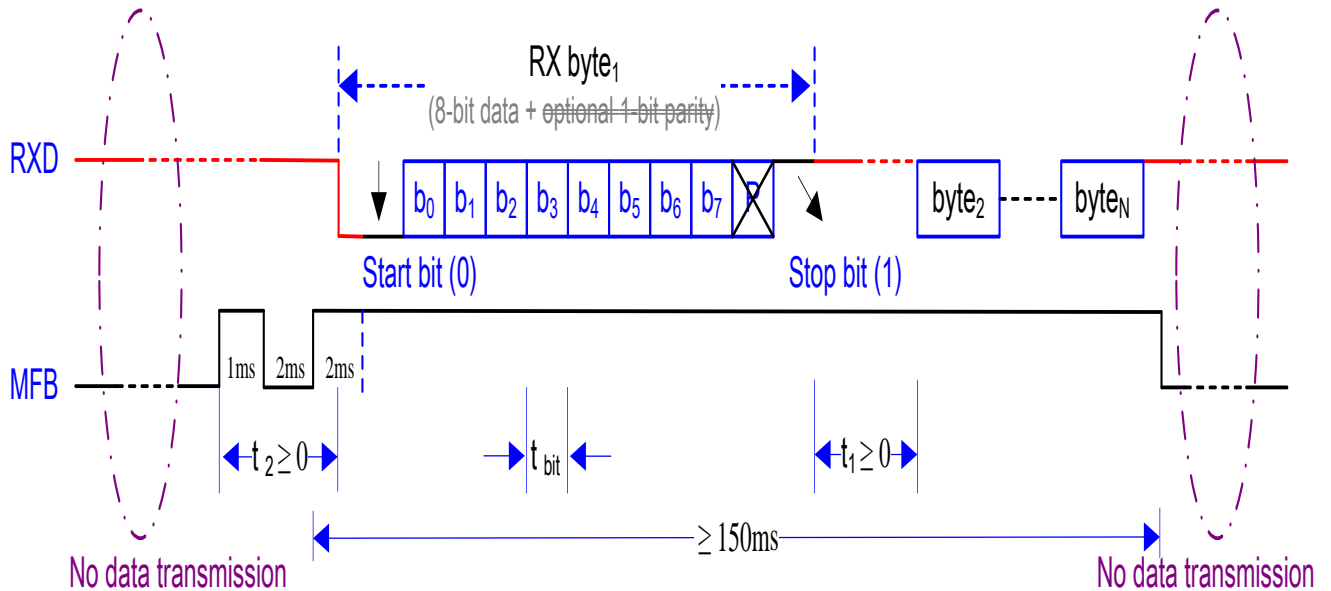
	ISSC SPK Module Pin definition	Note
TXD	<b>HCI_TXD</b>	
CTS	<b>PO_0</b>	SPK Module use to wakeup MCU and indicate TX action
RXD	<b>HCI_RXD</b>	
RTS	<b>MFB</b>	MCU use to wakeup SPK Module and indicate TX action

System Block Diagram:

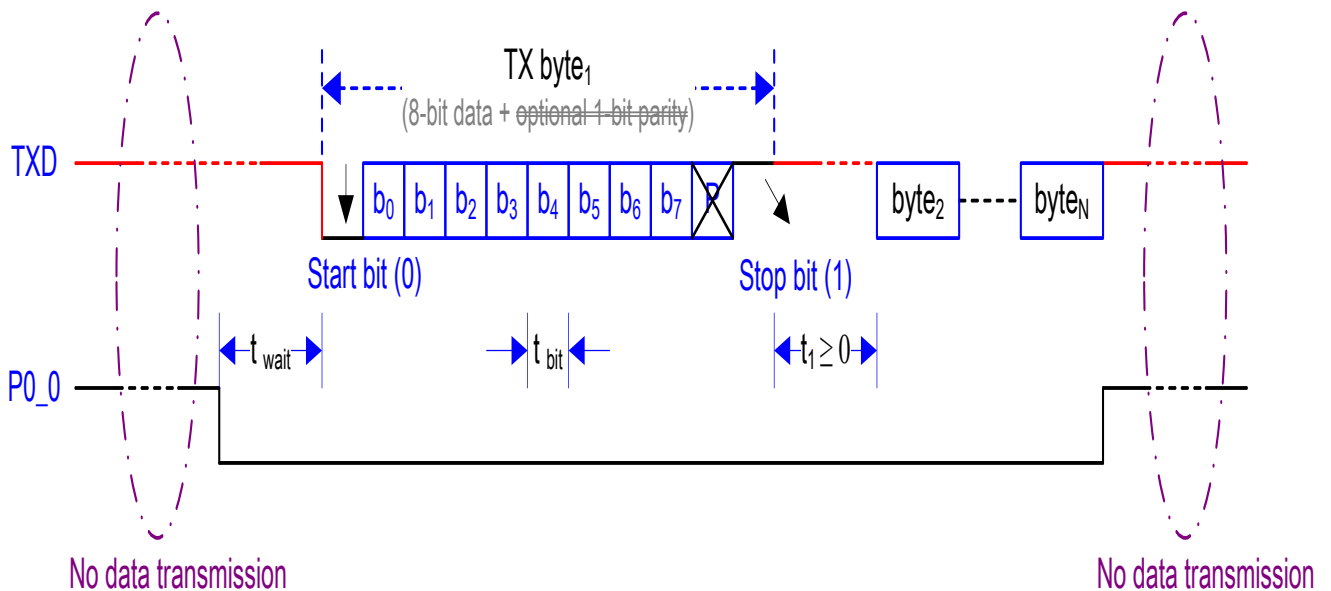


**UART Data Transfers:**

**MCU to SPK Module (Command):**



**SPK Module to MCU (Event):**

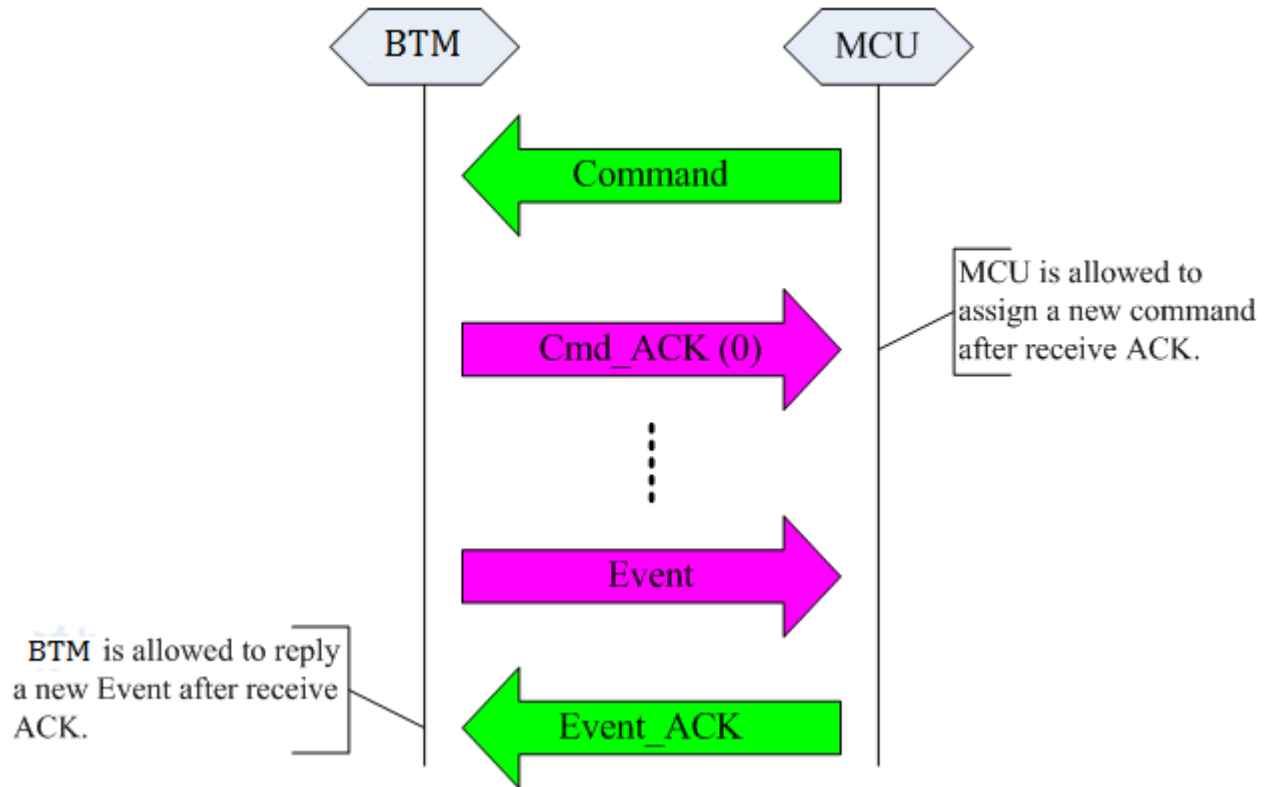


**Timing specifications**

Parameters	Minimum	Typical	Maximum	Unit	
$t_1$	0		$\infty$	us	
$t_2$	0	2	$\infty$	us	
$t_{wait}$		10		ms	
Baud rate ( $t_{bit}^{-1}$ )	16MHz clock	2,400	115,200	230,400	bps
	48MHz clock	2,400	115,200	3,000,000	bps

UART Protocol:

◎Command/Event Hand Shaking



◎Packet Format:

	SINC WORD	LENGTH	OP Code	OP Code PARAM	CHK SUM
BYTE NO	0	1(high byte) , 2(low byte)	3	3 ~ XX	Length + 3
SIZE (BYTE)	1	2	1	0~	1
VALUE	0xAA	1~65535	COMM	DATA	CHK SUM
		Check sum to be calculated			
		TARGET LENGTH			

※CHK SUM must be add to ( LENGTH + OP Code + OP Code PARAM ) equal to 0x00.

**MCU to SPK Module (Command):**

MCU send UART Command to SPK Module for specific Hands-Free 、A2DP or Mobile Phone Book Access purpose. SPK Module shall response the corresponded ACK event except the ACK event had been disable by Report Mask Command setting.

**Command Packet Format:**

Byte	Value	Meaning
0	0xAA	Start
1	0xNN	Payload length high byte(including command id and data)
2	0xNN	Payload length low byte(including command id and data)
3	0xNN	Command Id
4...	0xNN	Command parameters
Last byte	0xNN	Packet payload checksum
PS: sum of all bytes from byte1 to last byte is 0x00		

**Command List:**

Item	Command	Command ID
1	Make_Call	0x00
2	Make_Extension_Call	0x01
3	MMI_Action	0x02
4	Event_Mask_Setting	0x03
5	Music Control	0x04
6	Change_Device_Name	0x05
7	Change_PIN_Code	0x06
8	BTM_Parameter_Setting	0x07
9	Read_BTM_Version	0x08
10	Get_PB_By_AT_Cmd	0x09
11	Vendor_AT_Command	0x0A
12	AVC_Specific_Cmd	0x0B
13	AVC_Group_Navigation	0x0C
14	Read_Link_Status	0x0D
15	Read_Paired_Device_Record	0x0E
16	Read_Local_BD_Address	0x0F
17	Read_Local_Device_Name	0x10
18	Set_Access_PB_Method	0x11
19	Send_SPP/iAP_Data	0x12
20	BTM_Utility_Function	0x13

21	Event_ACK	0x14
22	Additional_Profiles_Link_Setup	0x15
23	Read_Linked_Device_Information	0x16
24	Profiles_Link_Back	0x17
25	Disconnect	0x18
26	MCU_Status_Indication	0x19
27	User_Confirm_SPP_Req_Reply	0x1A
28	Set_HF_Gain_Level	0x1B
29	EQ_Mode_Setting	0x1C
30	DSP_NR_CTRL	0x1D
31	GPIO_Control	0x1E
32	MCU_UART_Rx_Buffer_Size	0x1F
33	Voice_Prompt_Cmd	0x20
34	MAP_REQUEST	0x21
35	Security_Bonding_Req	0x22
36	Set_Overall_Gain	0x23

*Command parameters:*

Ex.

Command Format:	Command	Command ID	Command Parameters	Return Parameters
	MMI_ACTION	0x02	data_base_index, action	ACK
<b>Description:</b>	MMI action			
<b>Command Parameters:</b>	Data_base_index			SIZE: 1 BYTE
	<b>Value</b>	<b>Parameter Description</b>		
	0x00	database 0 that related to a dedicate HF device		
	0x01	database 1 that related to a dedicate HF device		
	<b>Action ID</b>			SIZE: 1 BYTE
	<b>Value</b>	<b>Parameter Description</b>		
	0x01	add/remove SCO link		
	0x02	force end active call		
	0x04	Accept an incoming call		
	0x05	Reject an incoming call		

Reference: [UART\\_CommandSet\\_v151\\_20140813.xlsx](#)

**SPK Module to MCU (Event):**

SPK Module reports the UART Event to update status while it changed. These reported Events can be disabled while “**Report Mask Command**” is assigned by MCU.

**Event Packet Format:**

Byte No.	Value	Meaning
0	0xAA	Start
1	0xNN	Payload length high byte(including command id and data)
2	0xNN	Payload length low byte(including command id and data)
3	0xNN	Event code
4...	0xNN	Event parameters
Last byte	0xNN	Packet payload checksum
PS: sum of all bytes from byte1 to last byte is 0x00		

**Event List:**

Item	Event	Event ID
1	Command_ACK	0x00
2	BTM_Status	0x01
3	Call_Status	0x02
4	Caller_ID	0x03
5	SMS_Received_Indication	0x04
6	Missed_Call_Indication	0x05
7	Phone_Max_Battery_Level	0x06
8	Phone_Current_Battery_Level	0x07
9	Roaming_Status	0x08
10	Phone_Max_Signal_Strength_Level	0x09
11	Phone_Current_Signal_Strength_Level	0x0A
12	Phone_Service_Status	0x0B
13	BTM_Battery_Status	0x0C
14	BTM_Charging_Status	0x0D
15	Reset_To_Default	0x0E
16	Report_HF_Gain_Level	0x0F
17	EQ_Mode_Indication	0x10
18	PBAP_Missed_Call_History	0x11
19	PBAP_Received_Call_History	0x12
20	PBAP_Dialed_Call_History	0x13
21	PBAP_Combine_Call_History	0x14

22	Phonebook_Contacts	0x15
23	PBAP_Access_Finish	0x16
24	Read_Linked_Device_Information_Reply	0x17
25	Read_BTM_Version_Reply	0x18
26	Call_List_Report	0x19
27	AVC_Specific_Rsp	0x1A
28	BTM_Utility_Req	0x1B
29	Vendor_AT_Cmd_Reply	0x1C
30	Report_Vendor_AT_Event	0x1D
31	Read_Link_Status_Reply	0x1E
32	Read_Paired_Device_Record_Reply	0x1F
33	Read_Local_BD_Address_Reply	0x20
34	Read_Local_Device_Name_Reply	0x21
35	Report_SPP/iAP_Data	0x22
36	Report_Link_Back_Status	0x23
37	Ringtone_Finish_Indicate	0x24
38	User_Confrim_SSP_Req	0x25
39	Report_AVRCP_Vol_Ctrl	0x26
40	Report_Input_Signal_Level	0x27
41	Report_iAP_Info	0x28
42	REPORT_AVRCP_ABS_VOL_CTRL	0x29
43	Report_Voice_Prompt_Status	0x2A
44	Report_MAP_Data	0x2B
45	Security_Bonding_Res	0x2C
46	Report_Type_Codec	0x2D

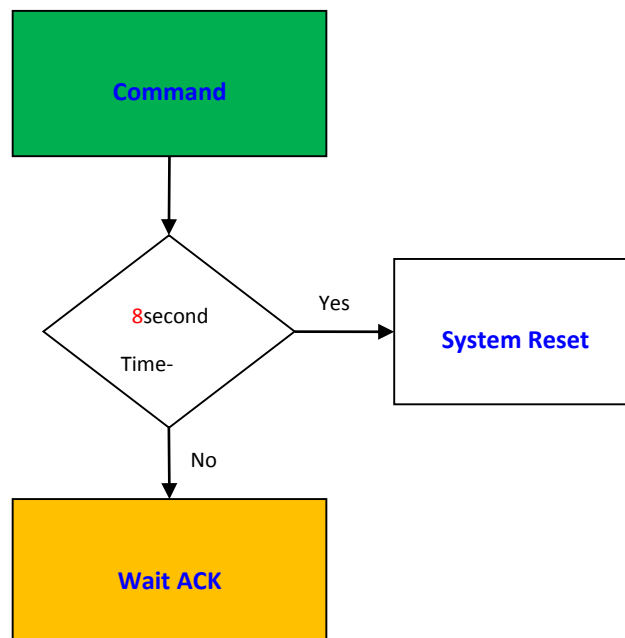
**Event parameters:**

Ex.

Event Format:	Event	Event Code	Event Parameters
	Call_Status	0x02	data_base_index, call_status
<b>Description:</b>	This event is used to indicate about the HF call status of BTM.		
<b>Event Parameters:</b>	data_base_index		SIZE: 1 BYTE
	<b>Value</b>	<b>Parameter Description</b>	
	0x00	database 0 for a dedicate link	
	0x01	database 1 for a dedicate link	
	call_status		SIZE: 1 BYTE
	<b>Value</b>	<b>Parameter Description</b>	
	0x00	Idle	
	0x01	voice dial	
	0x02	incoming call	
	0x03	outgoing call	
	0x04	call active	
	0x05	a call active with a call waiting	
	0x06	a call active with a call hold	

**Time-Out Mechanism:**

Every command accepts, then MCU wait SPK module response event. If SPK module has occurs unknown cause no answer, MCU need a time-out mechanism to “RESET” system restart.

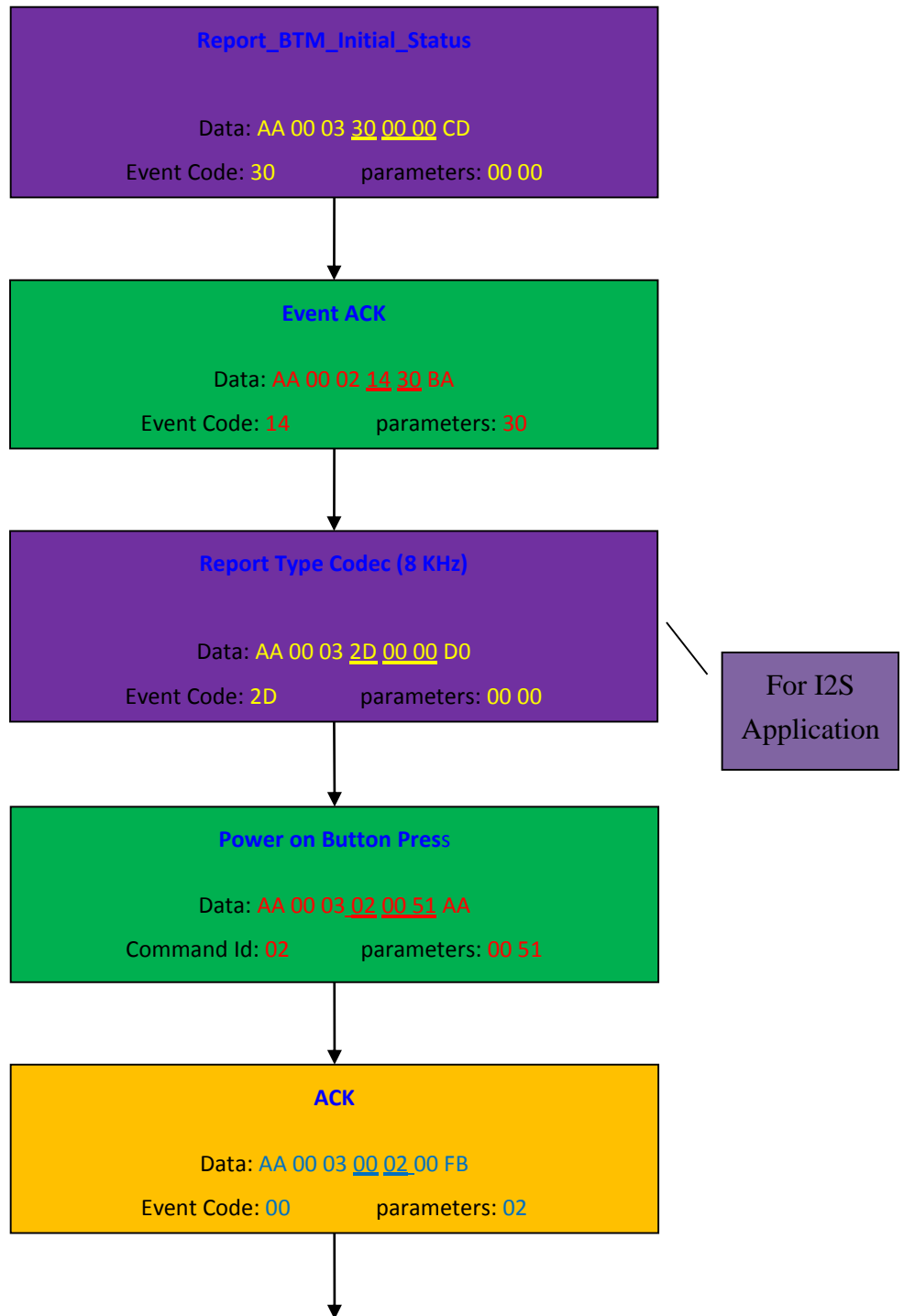


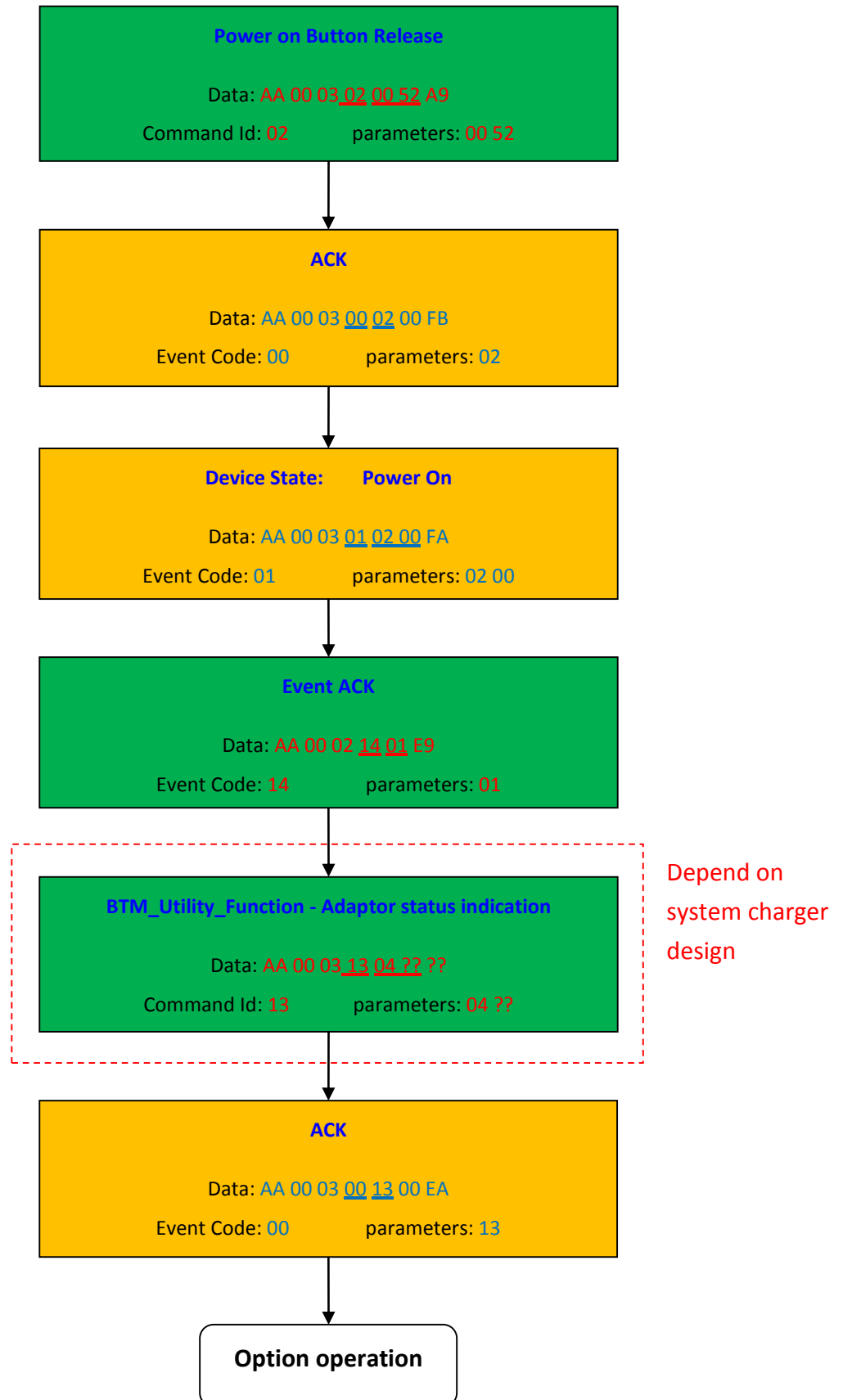
※ ISSC recommend the MCU at 8s time-out then retry send command 2<sup>th</sup>~3<sup>rd</sup> again. If SPK module has no any response, then force system reset.

Procedure:

- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

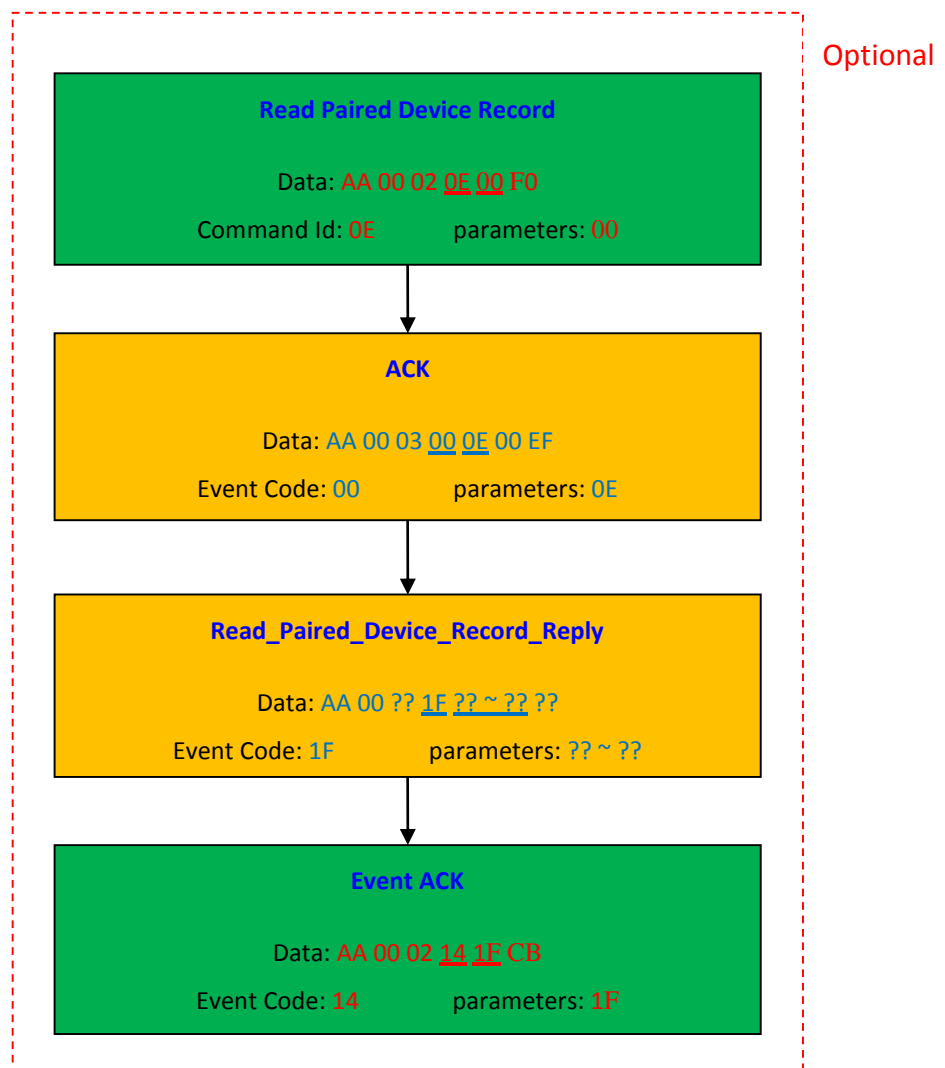
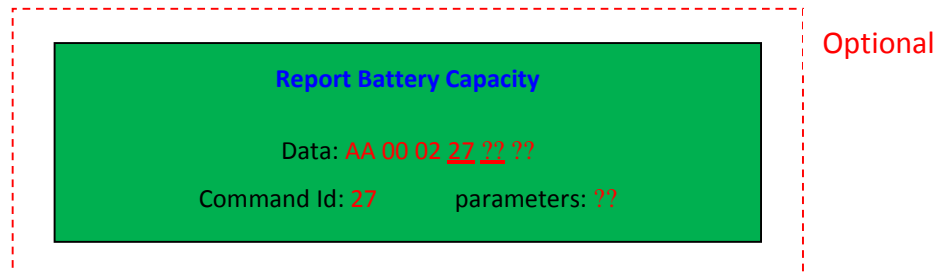
**Power On (UI Tool configure "MFB Power ON/OFF")**

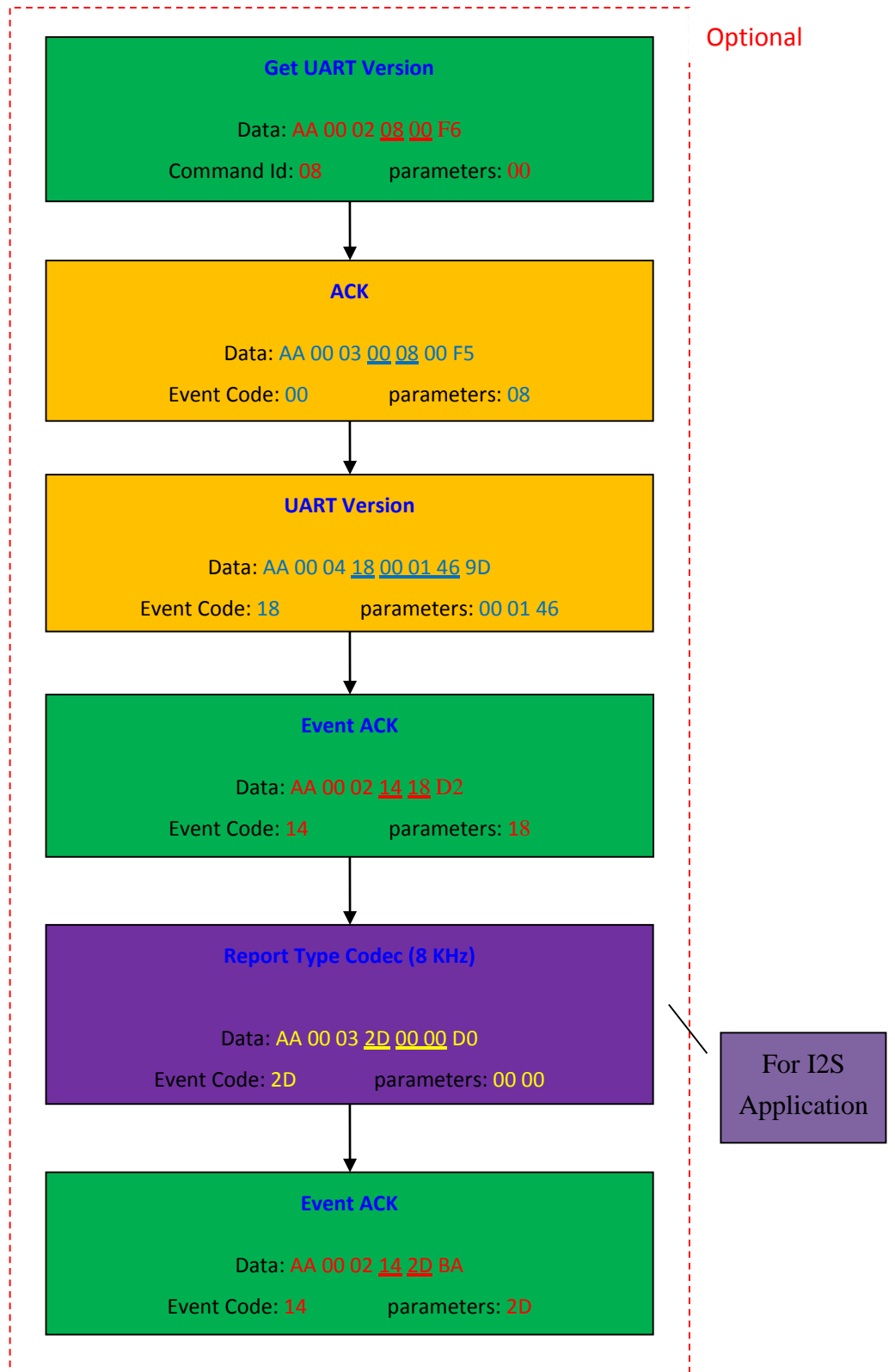


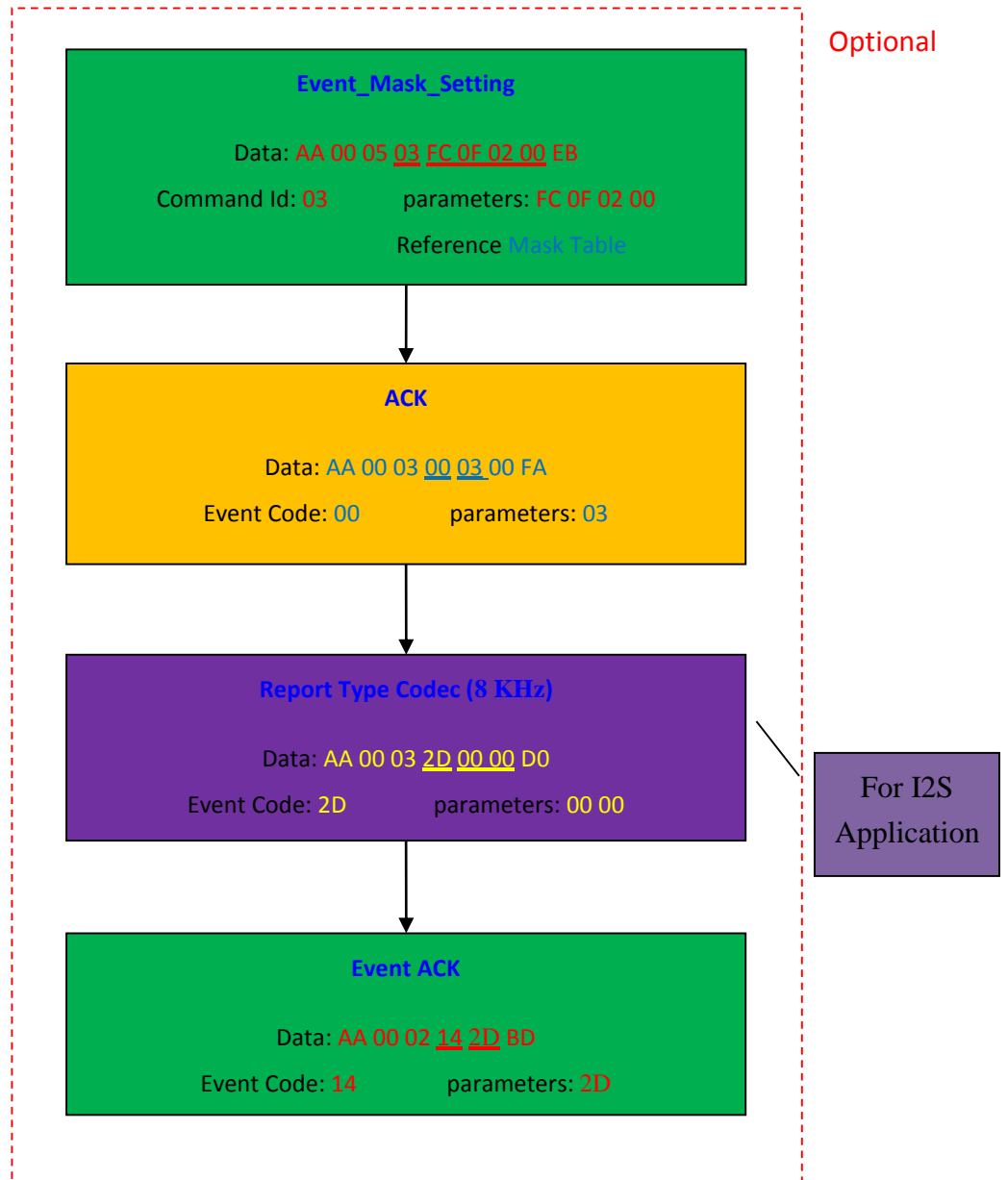


- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

**Optional operation:**

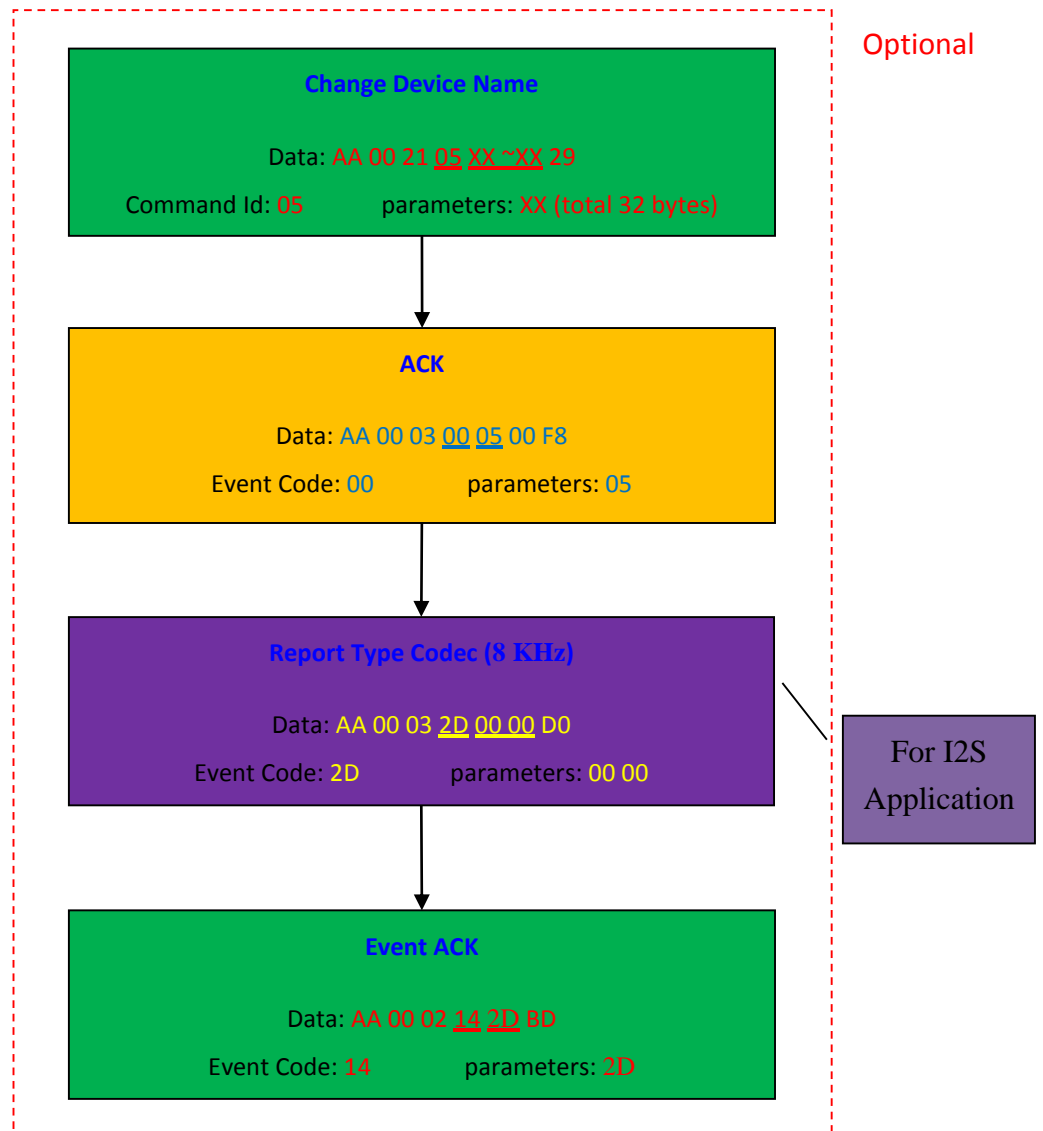




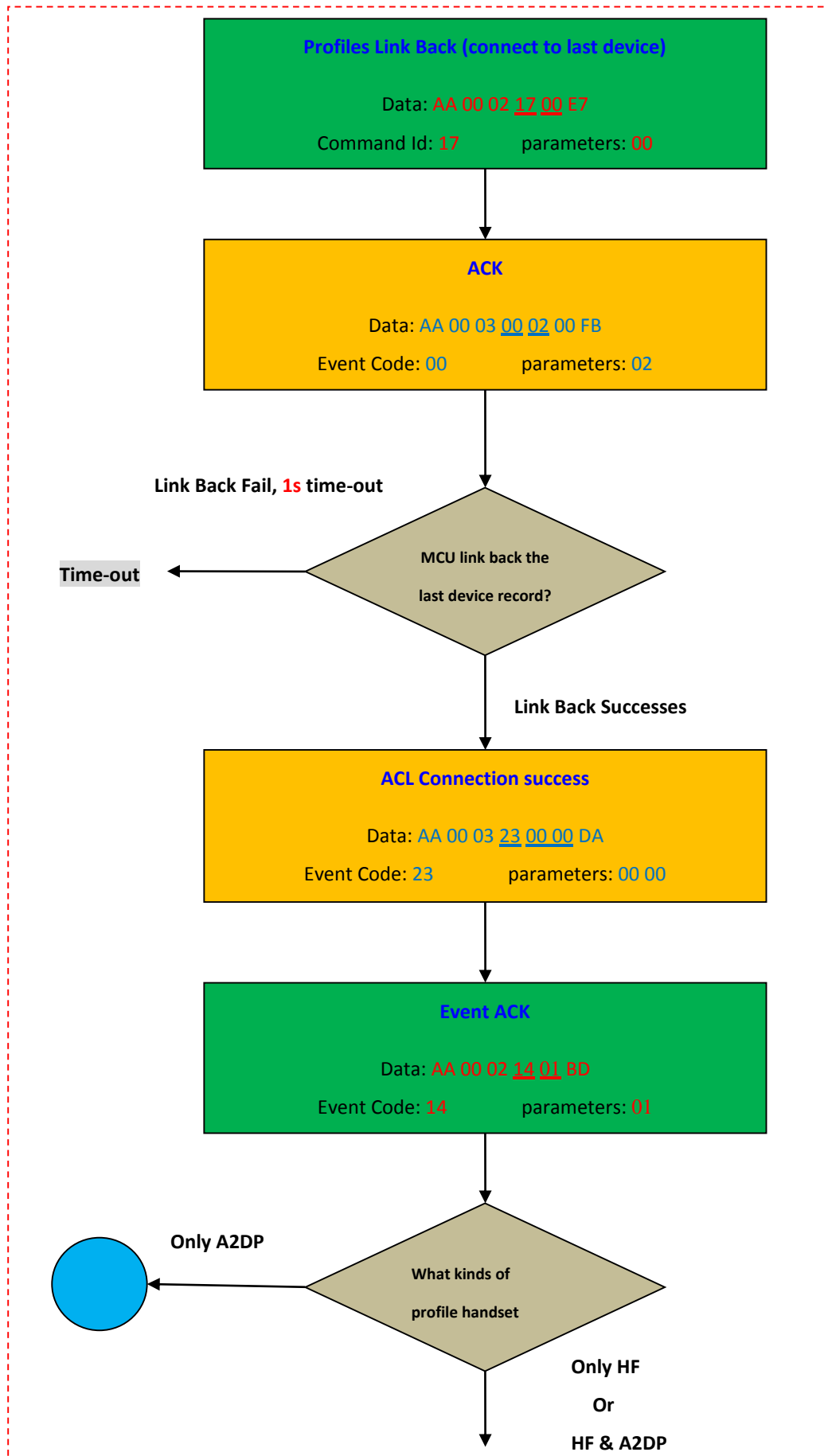


***Mask Table***

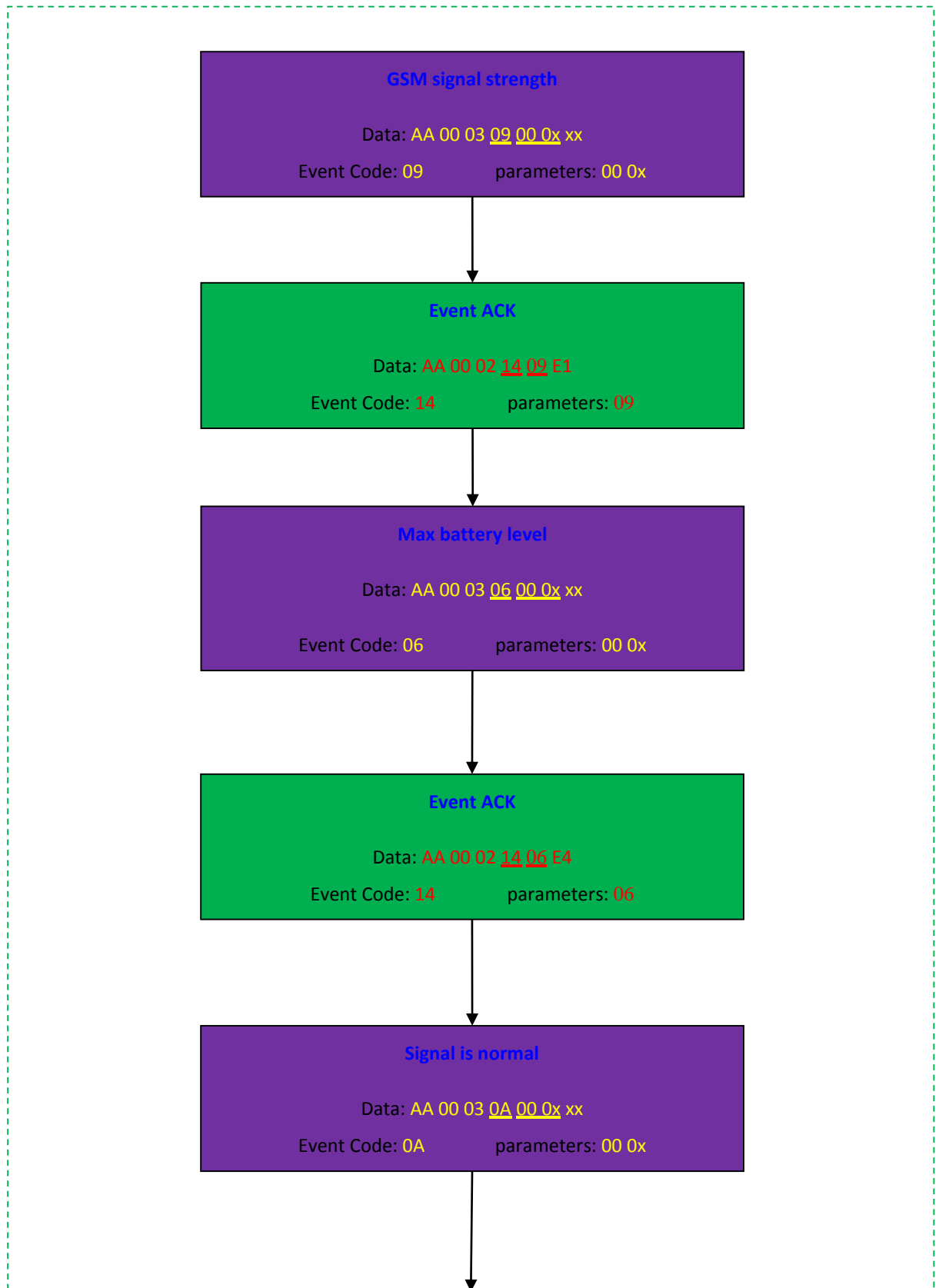
Ex. Byte 0 = 0xF8 => Bit7~3 = 1: Mask no display state  
 => Bit2~0 = 0: Display **Call Status** 、 **SPK Module Status** 、 **ACK Event**

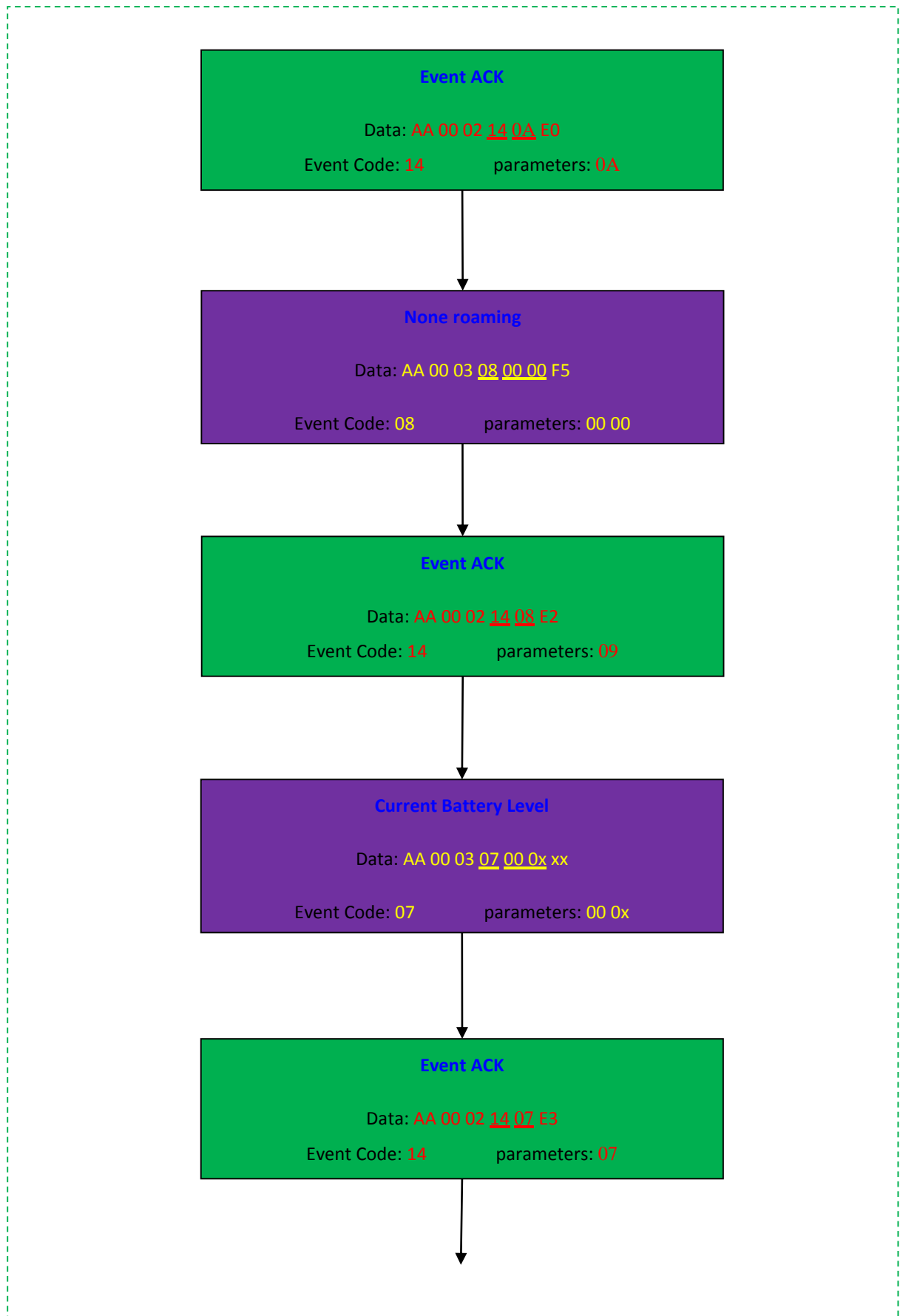


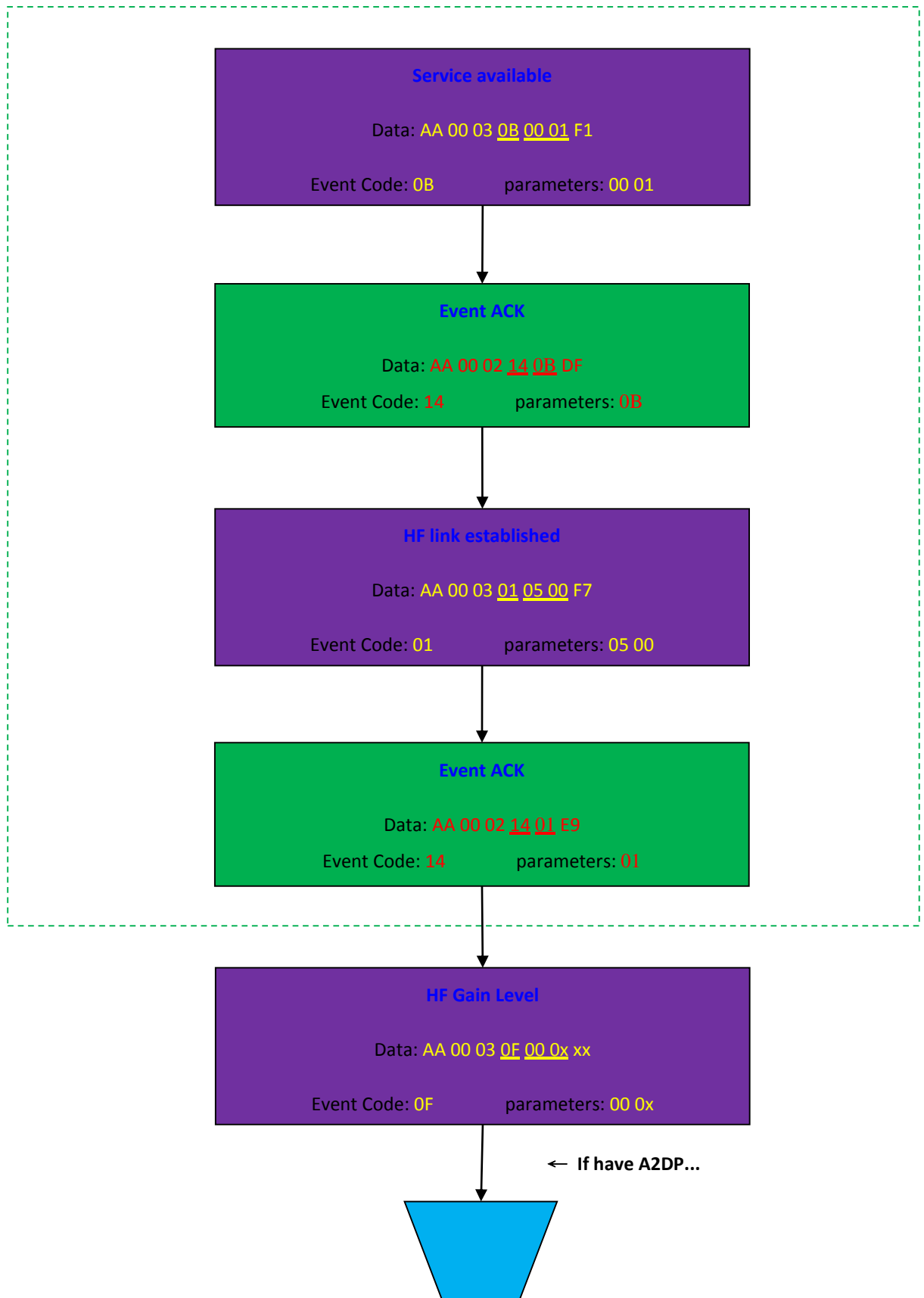
Optional






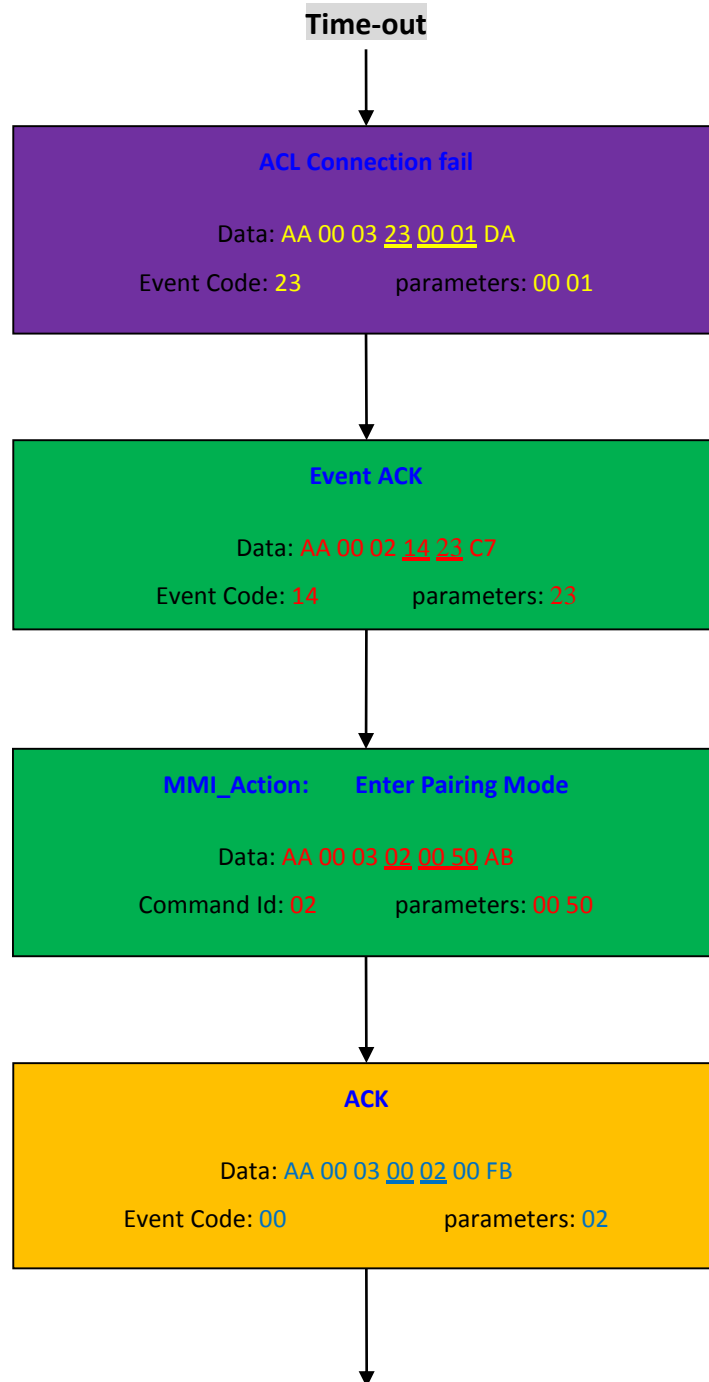
Depend on Mask table to show. ↓

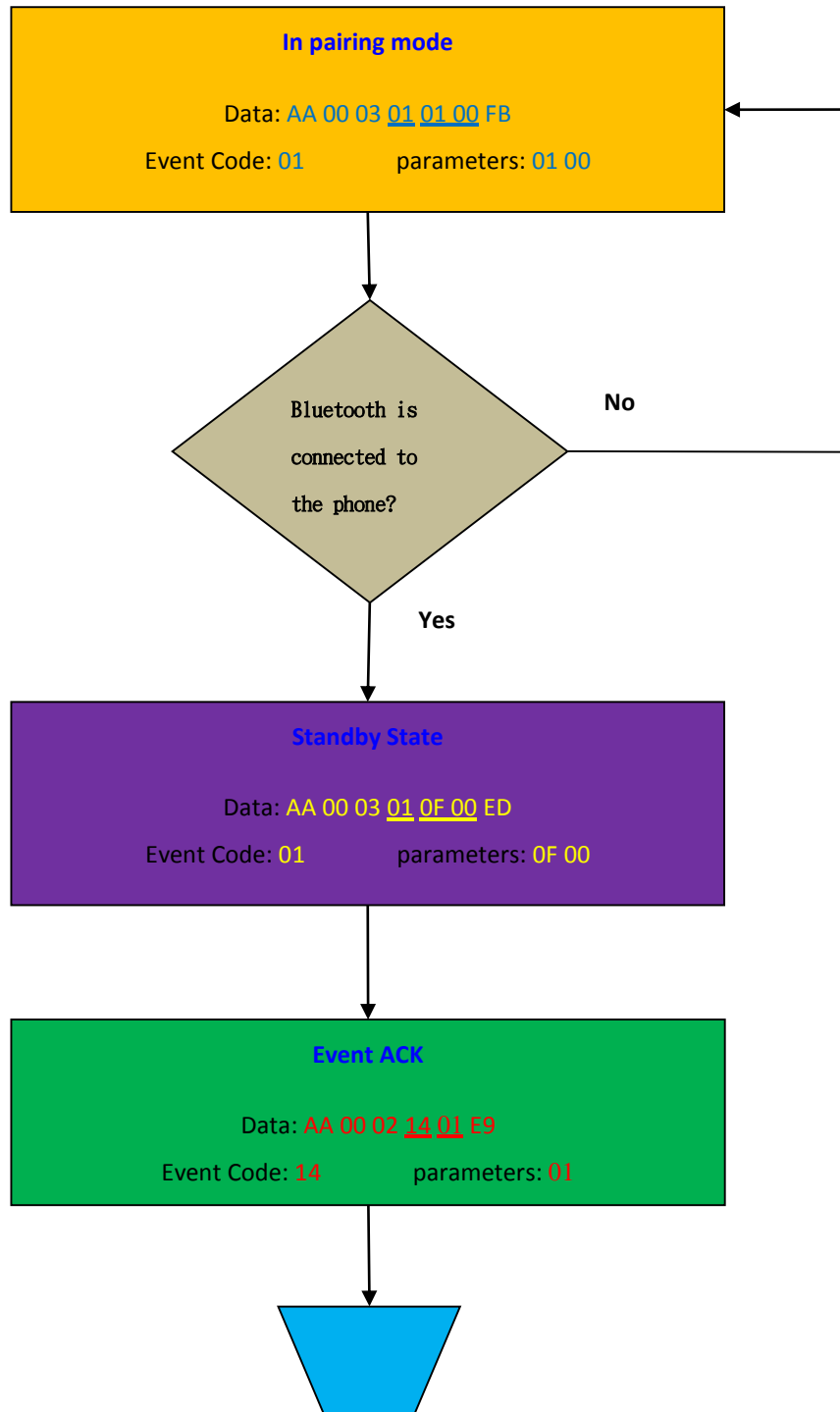







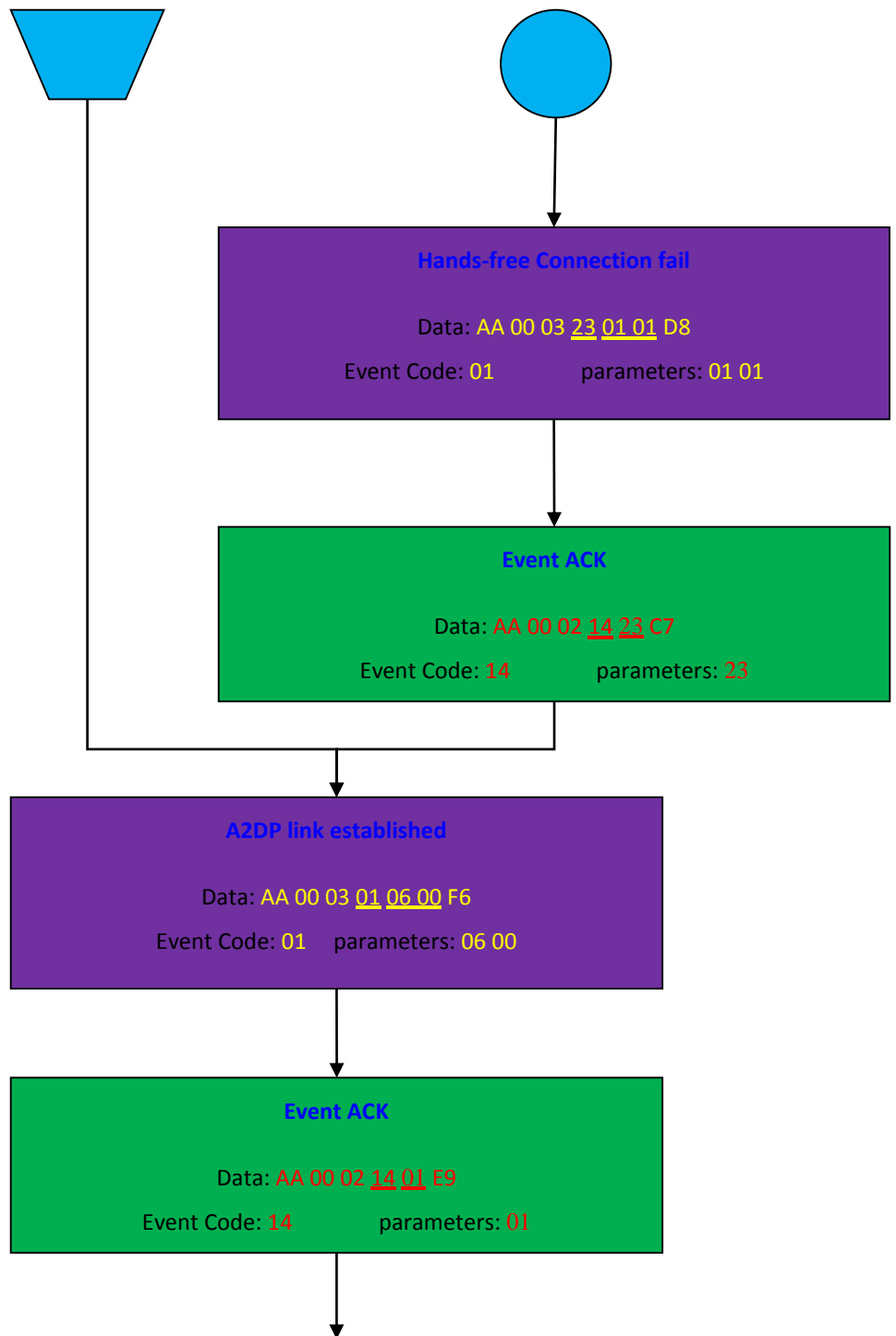


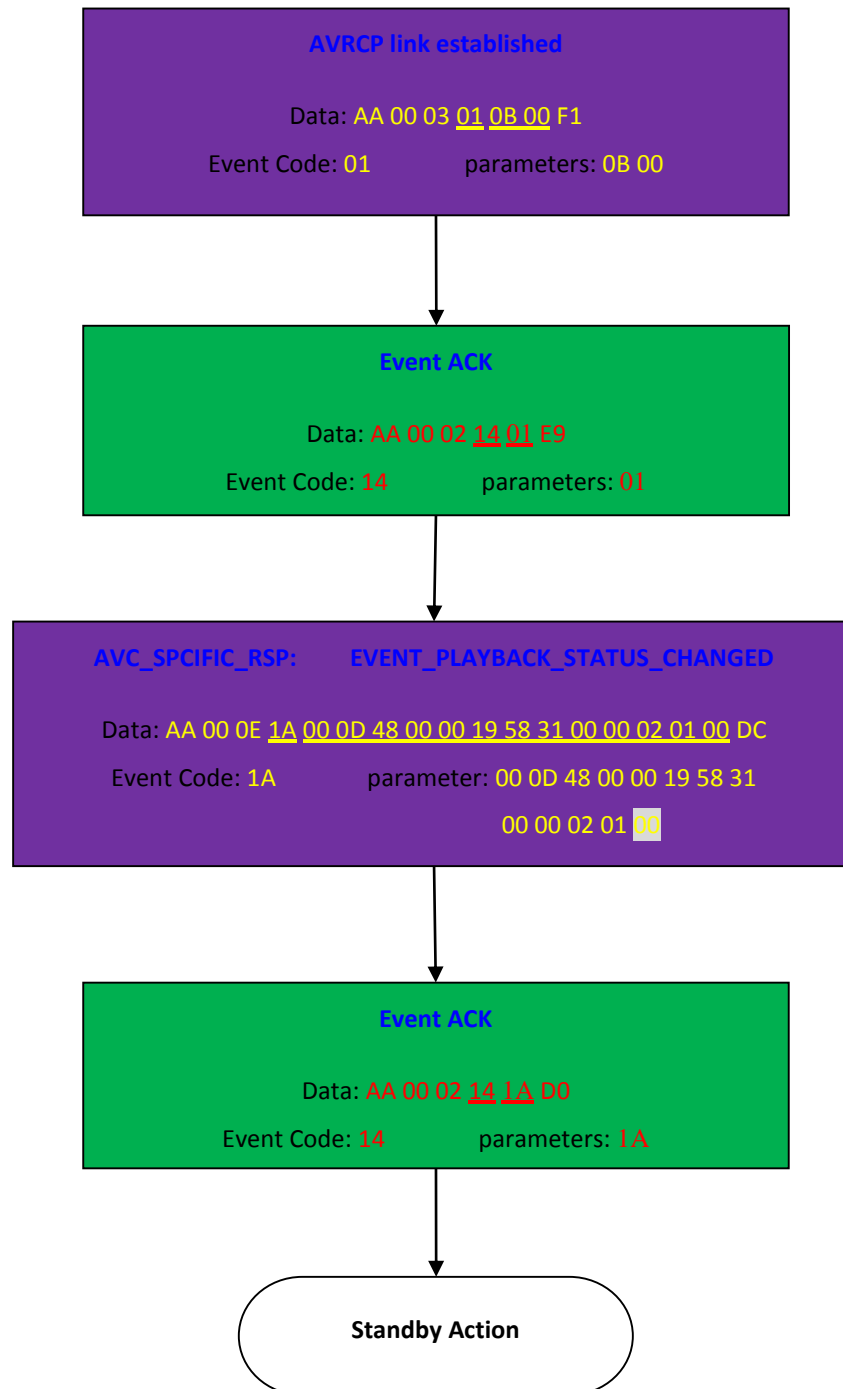
-  : MCU → SPK Module       : SPK Module → MCU (Command Accept)
-  : SPK Module → MCU (No Command Action)





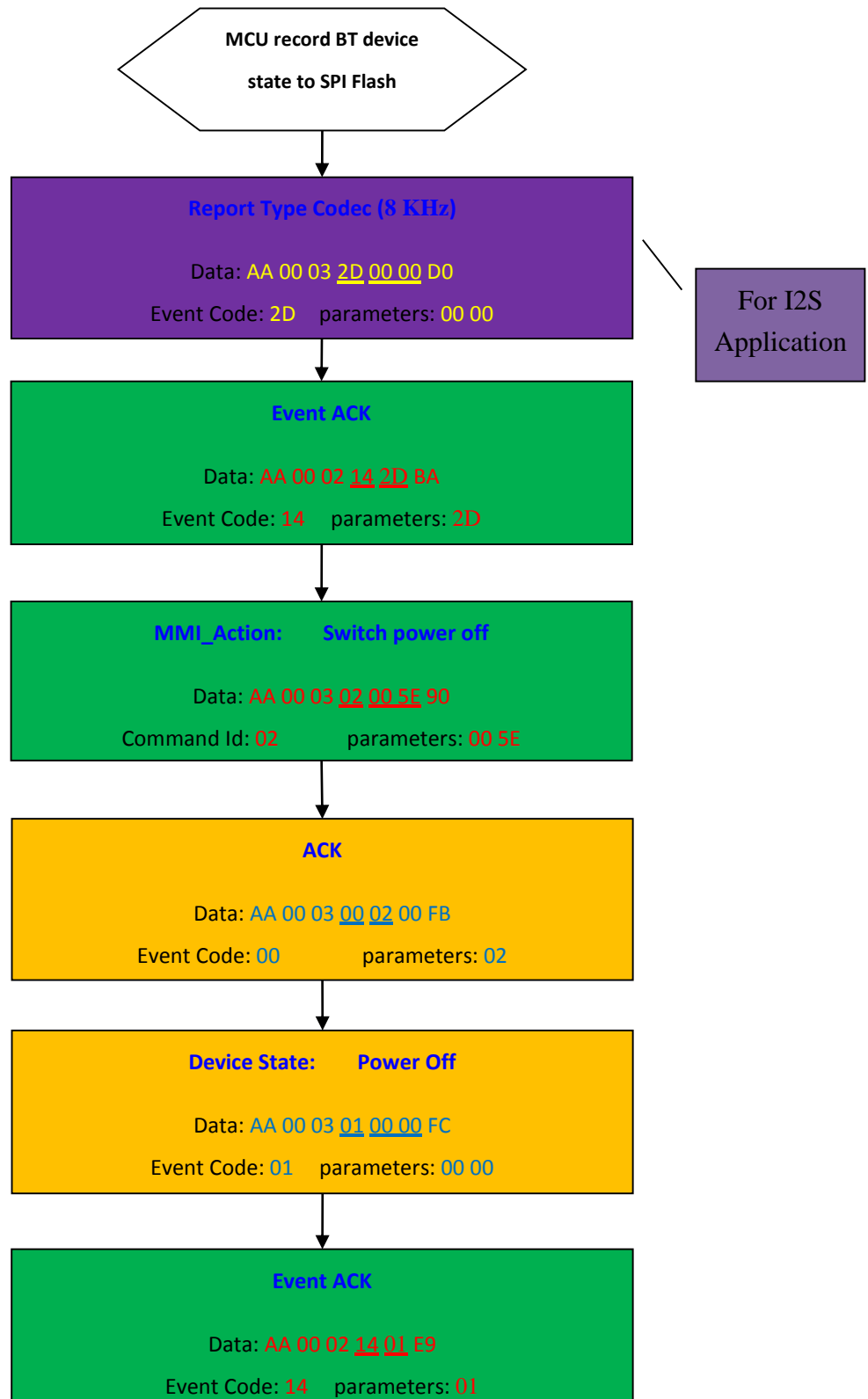
-  : MCU → SPK Module
-  : SPK Module → MCU (Command Accept)
-  : SPK Module → MCU (No Command Action)






- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

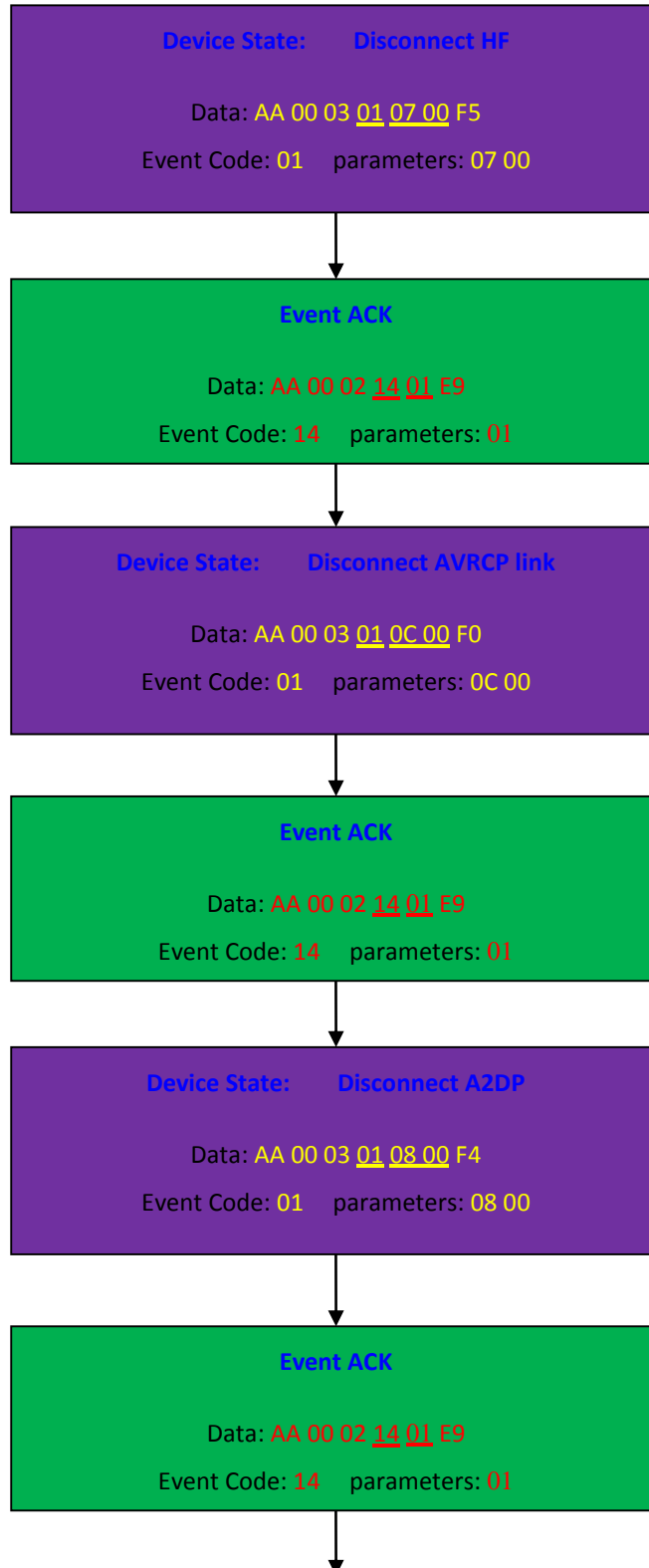
**(Soft) Power Off**

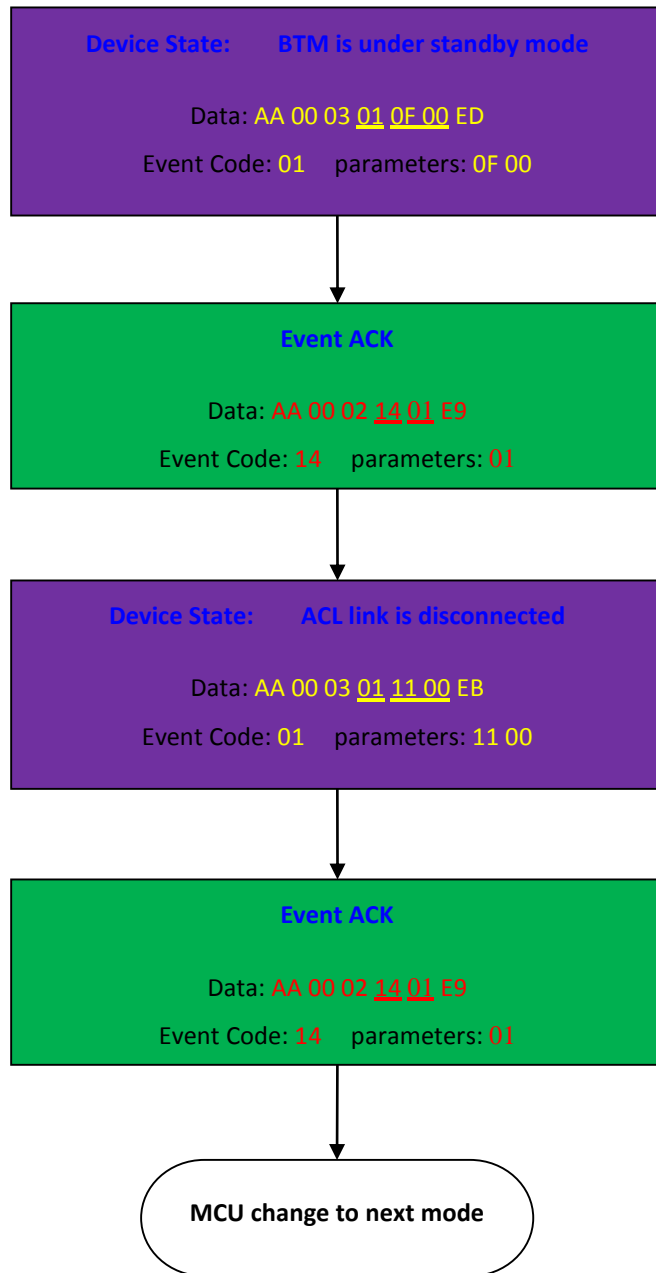


 : MCU → SPK Module

 : SPK Module → MCU (No Command Action)

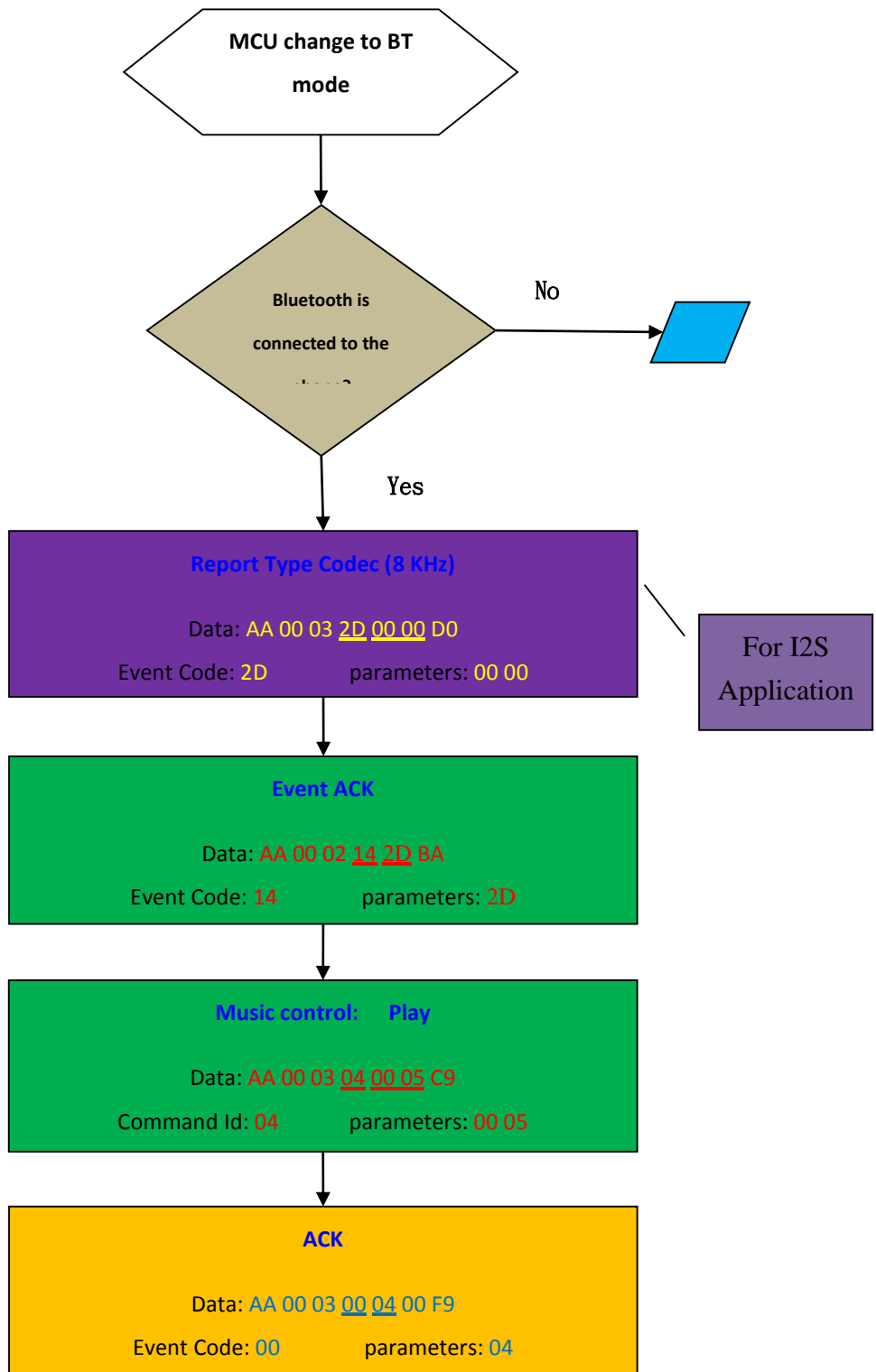
**Disconnected (*Hands-free operate*)**





- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

**Enter Pairing Mode**



**AVC\_SPCIFIC\_RSP: EVENT\_PLAYBACK\_STATUS\_CHANGED**

Data: AA 00 0E 1A 00 0D 48 00 00 19 58 31 00 00 02 01 01 DD

Event Code: 1A parameter: 00 0D 48 00 00 19 58 31  
00 00 02 01 01

**Event ACK**

Data: AA 00 02 14 1A D0

Event Code: 14 parameters: 1A

**AVC\_SPCIFIC\_RSP: EVENT\_PLAYBACK\_STATUS\_CHANGED**

Data: AA 00 0D 1A 00 0F 48 00 00 19 58 31 00 00 01 01 DB

Event Code: 1A parameter: 00 0F 48 00 00 19 58 31  
00 00 01 01

**Event ACK**

Data: AA 00 02 14 1A D0

Event Code: 14 parameters: 1A

**Report Type Codec (441.1 KHz or 48 KHz)**

Data: AA 00 03 2D 0x00 xx

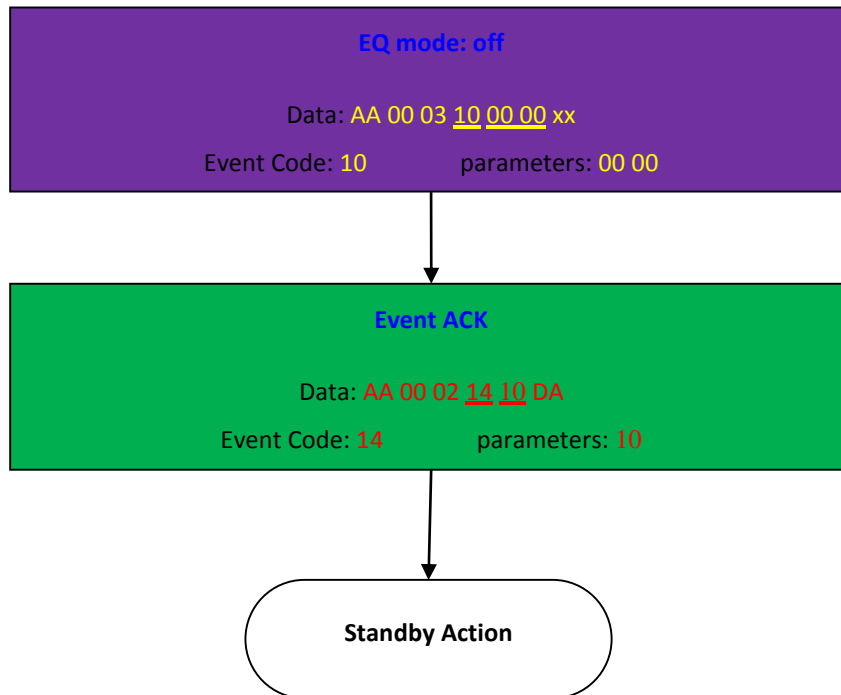
Event Code: 2D parameters: 0x00

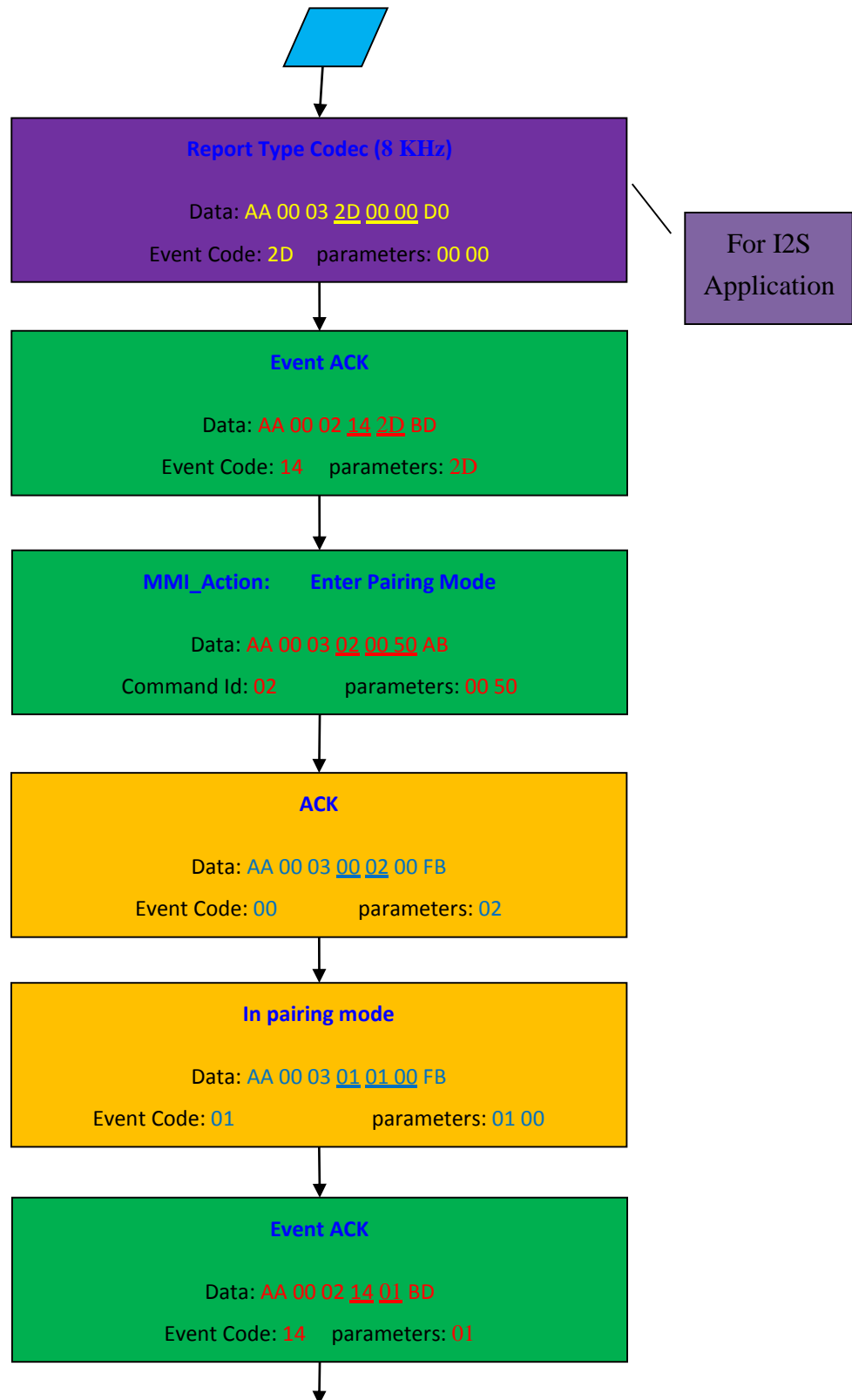
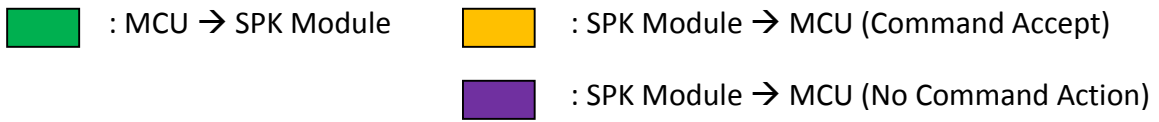
For I2S Application

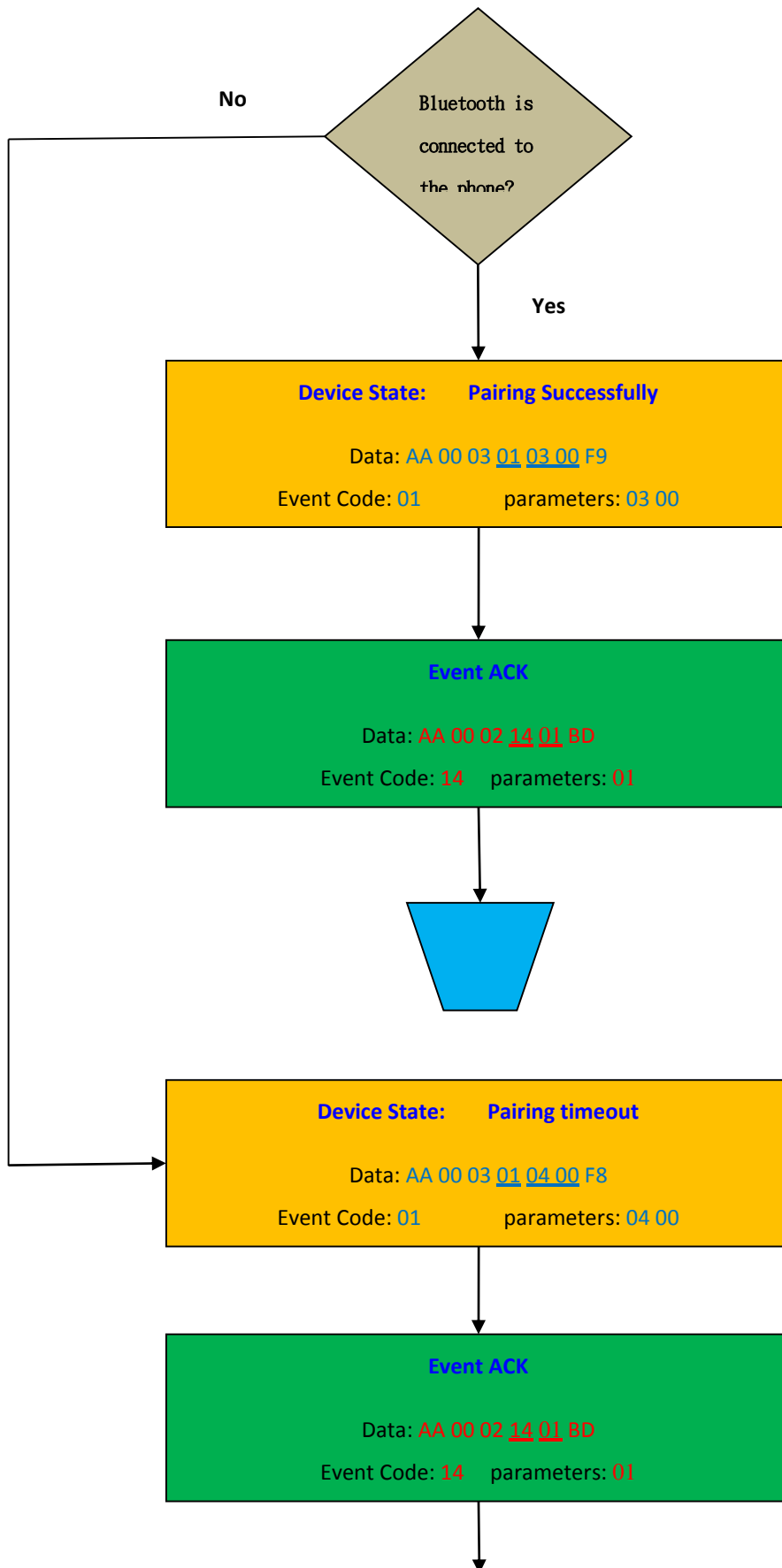
**Event ACK**

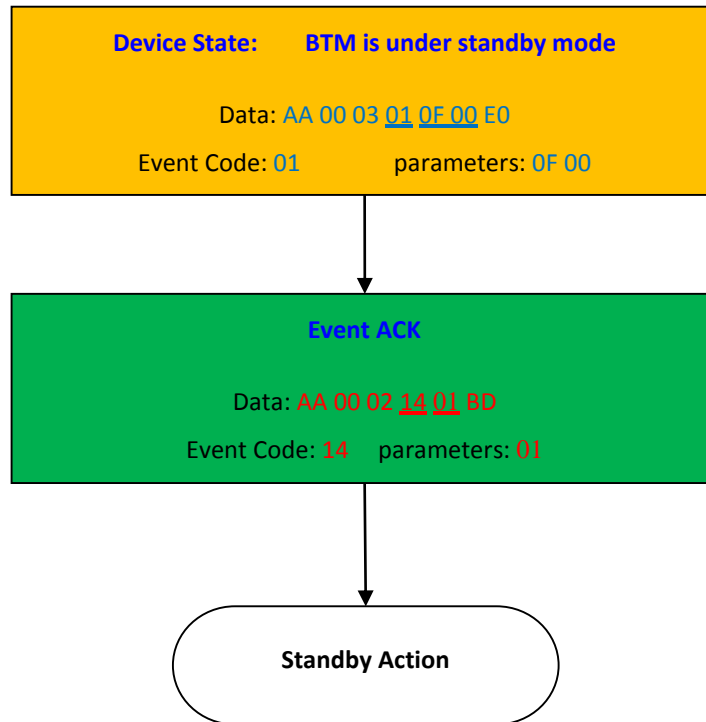
Data: AA 00 02 14 2D BD

Event Code: 14 parameters: 2D





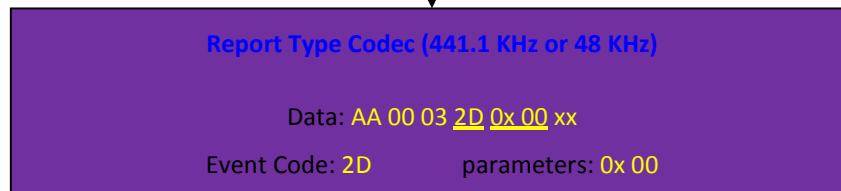
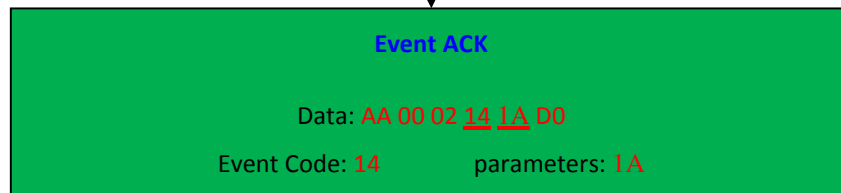
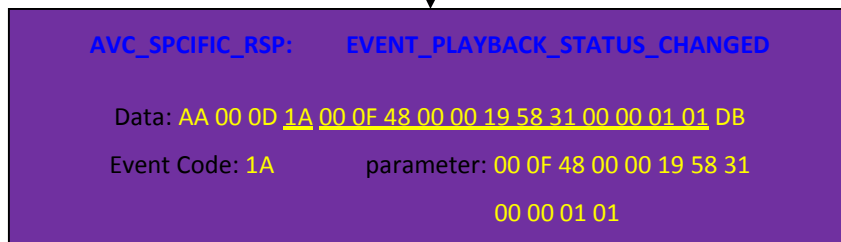
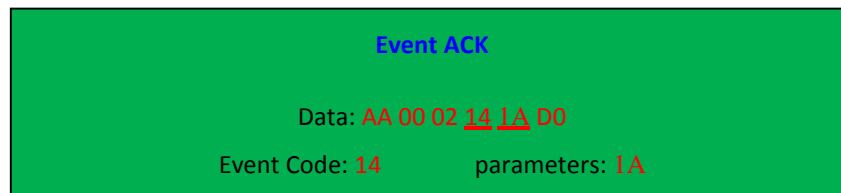




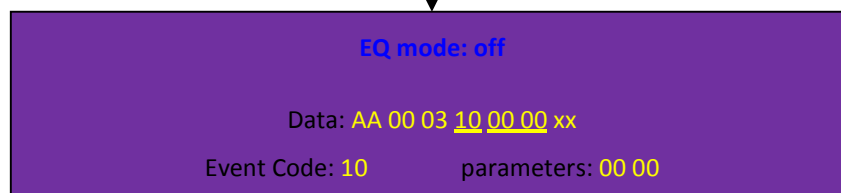
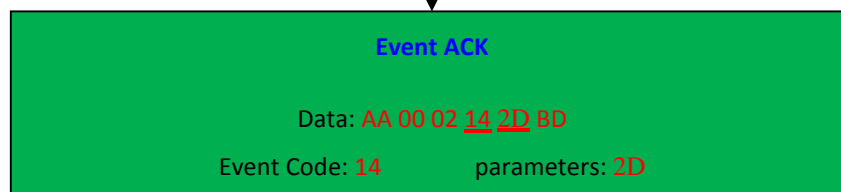
- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

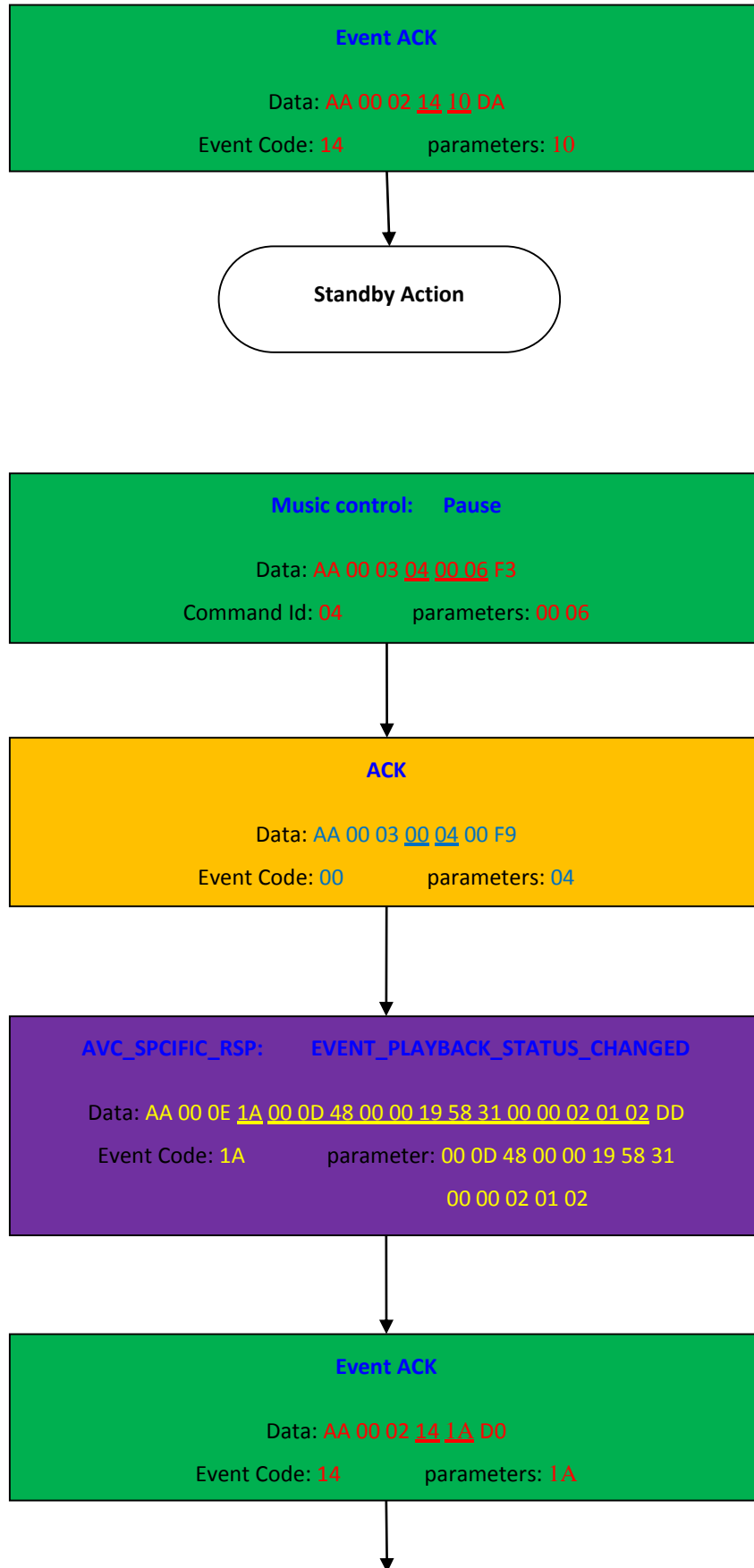
**Play/Pause Music (@BT Mode)**

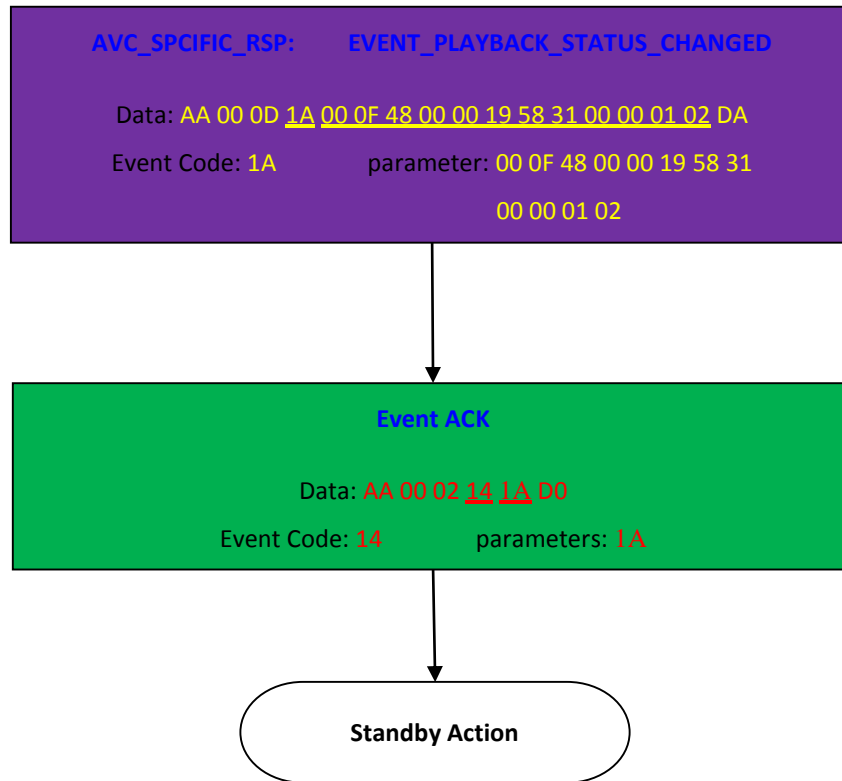




For I2S  
Application

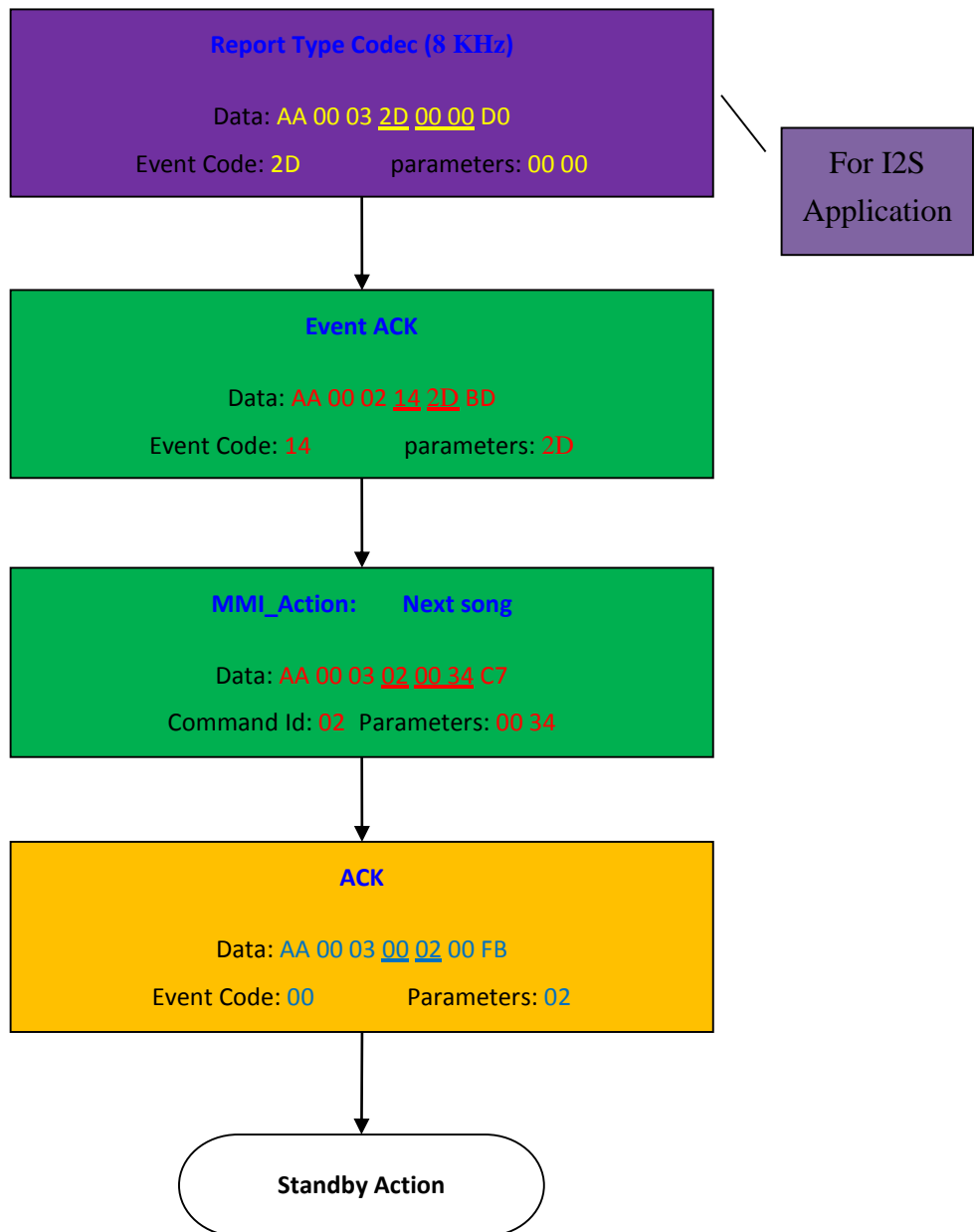


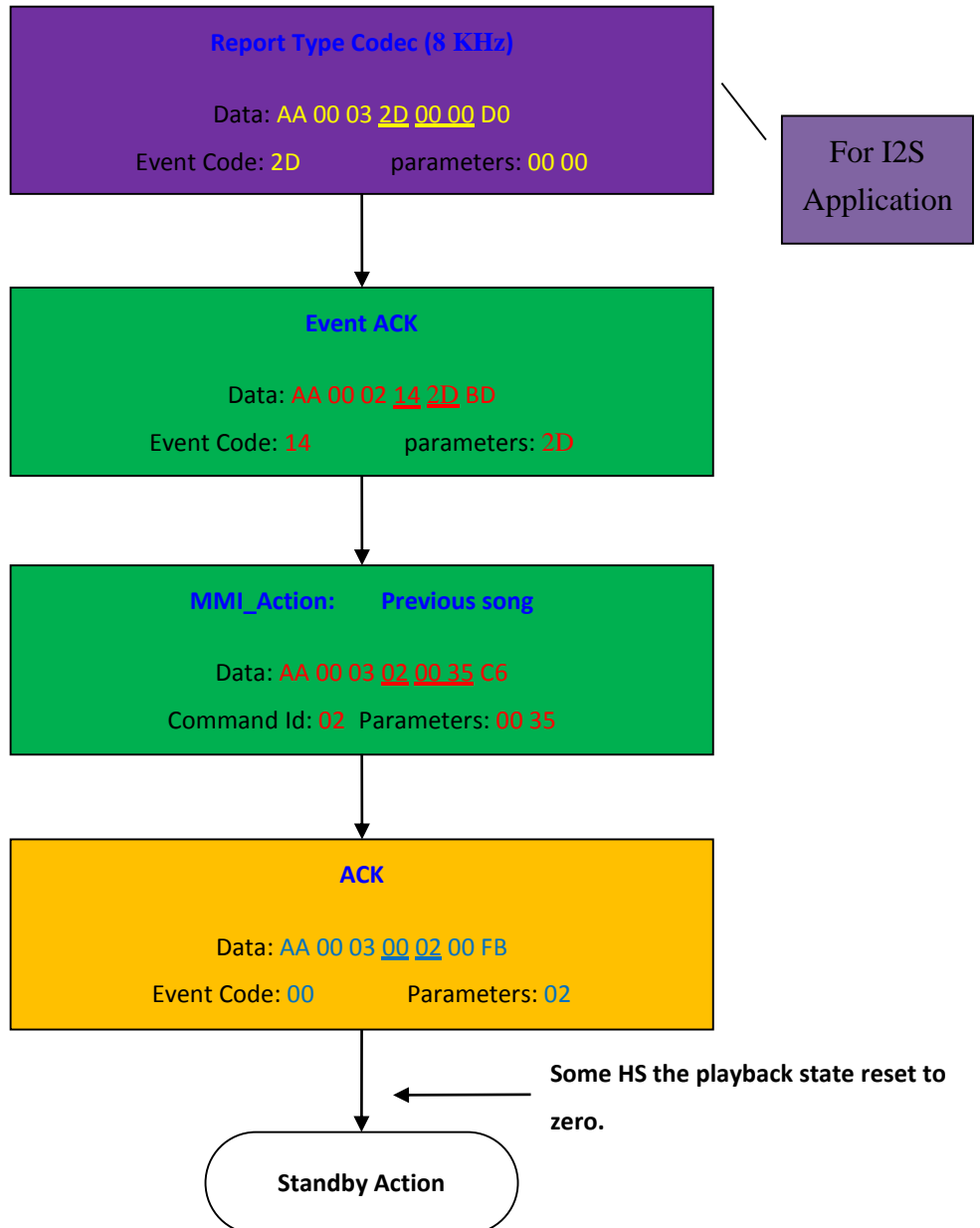




- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

**Next/Previous song (@BT Mode)**

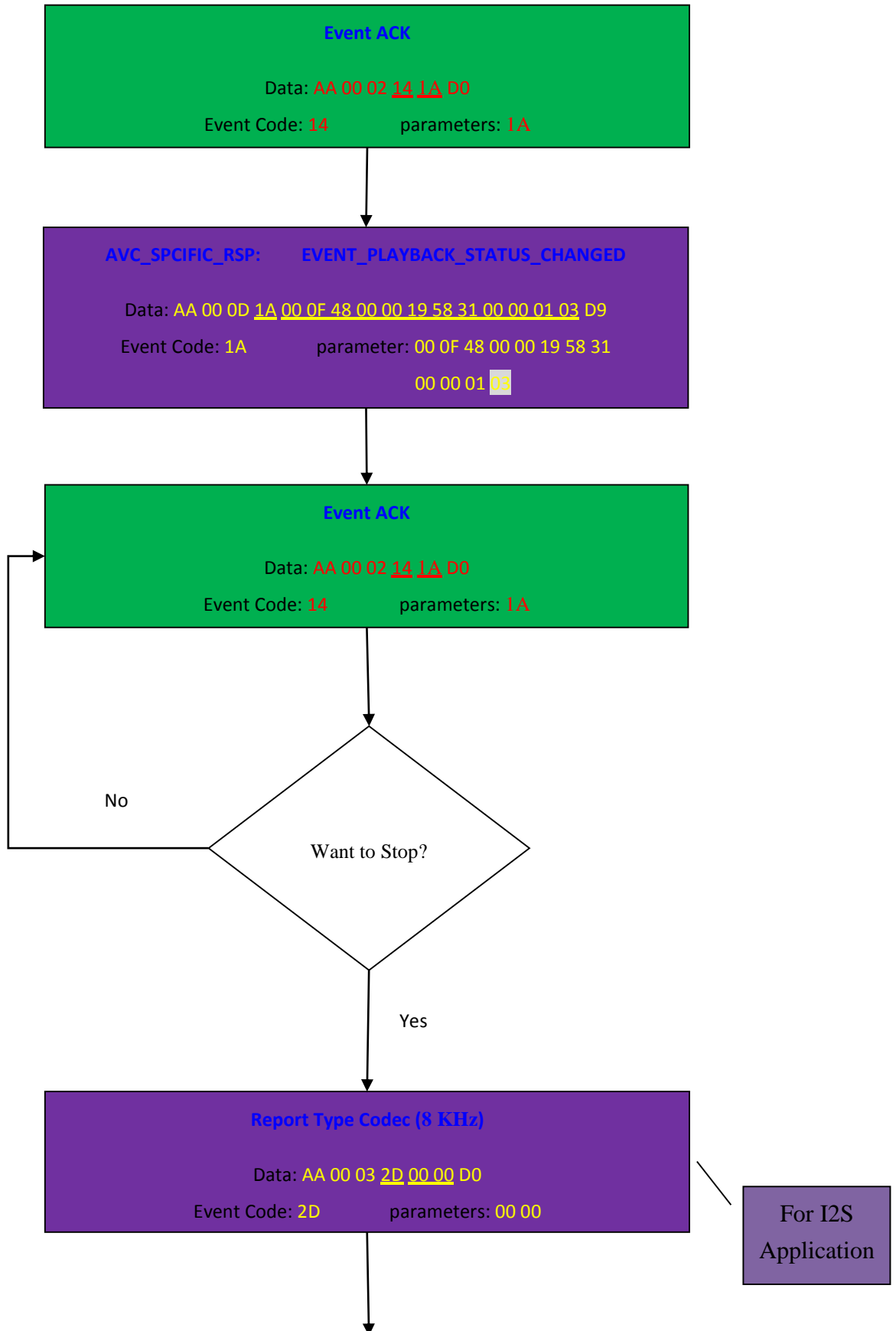


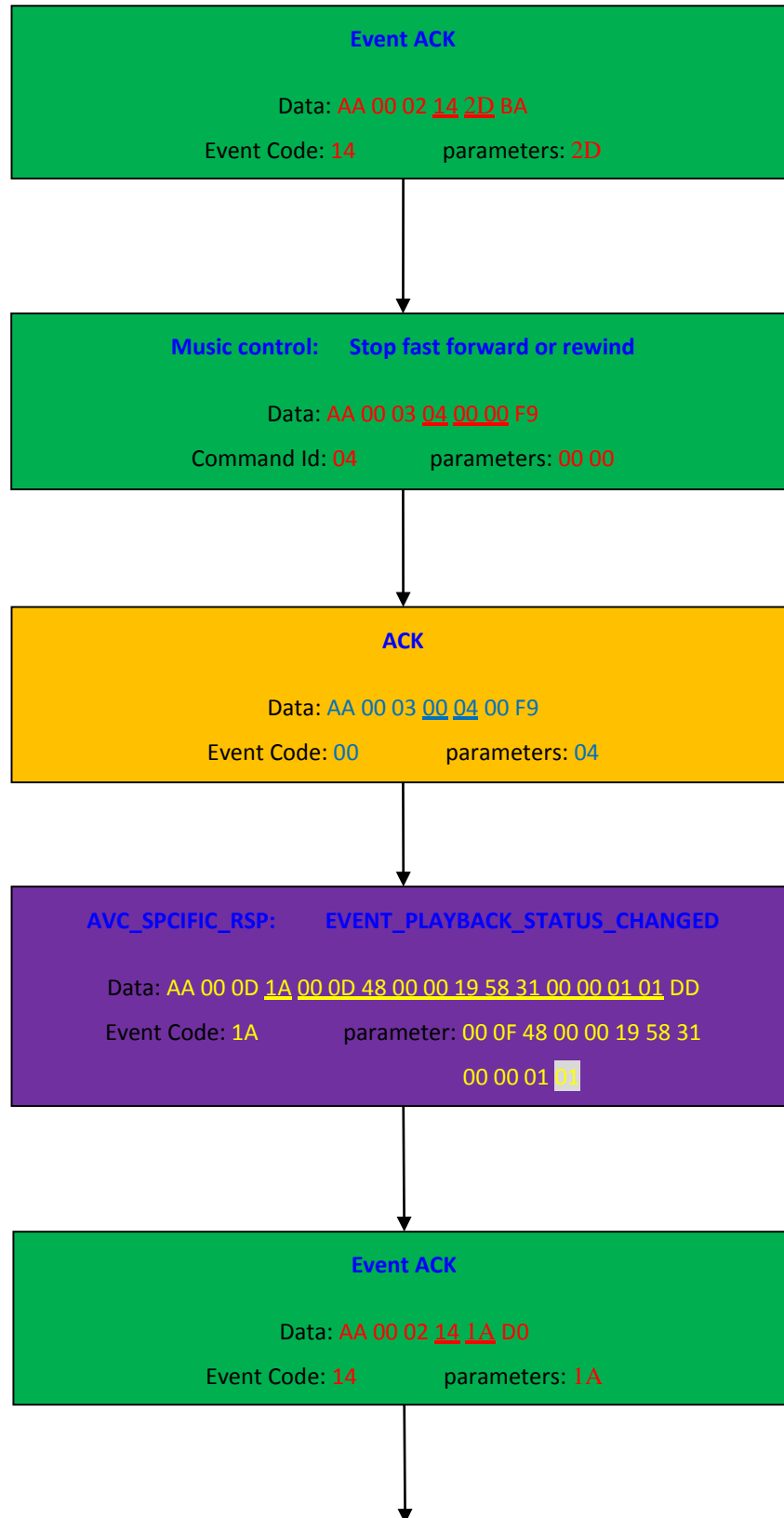


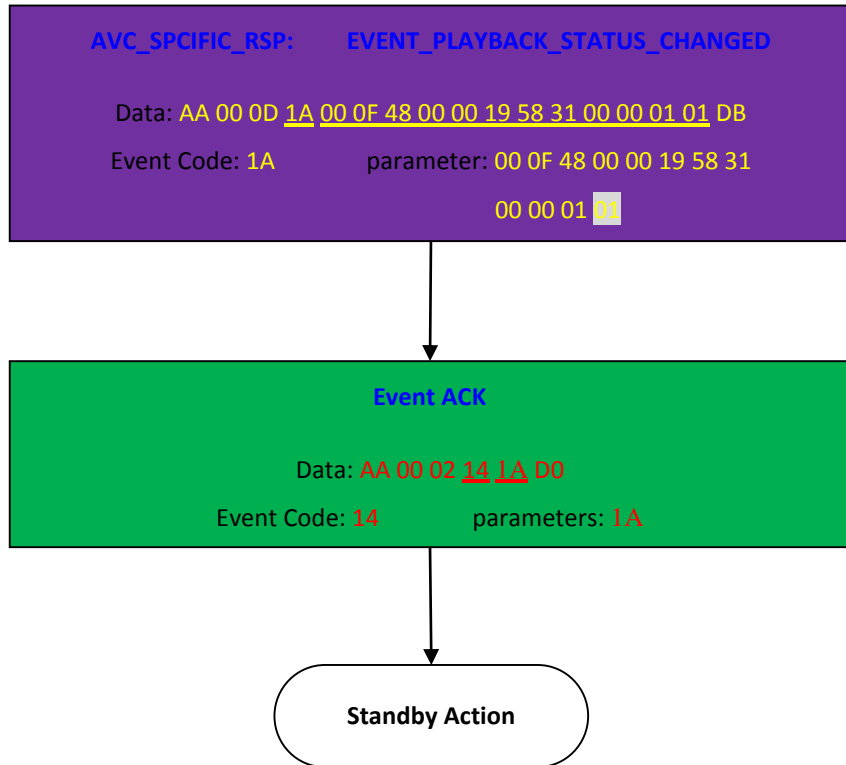
- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

**Fast forward/Rewind (@BT Mode)**







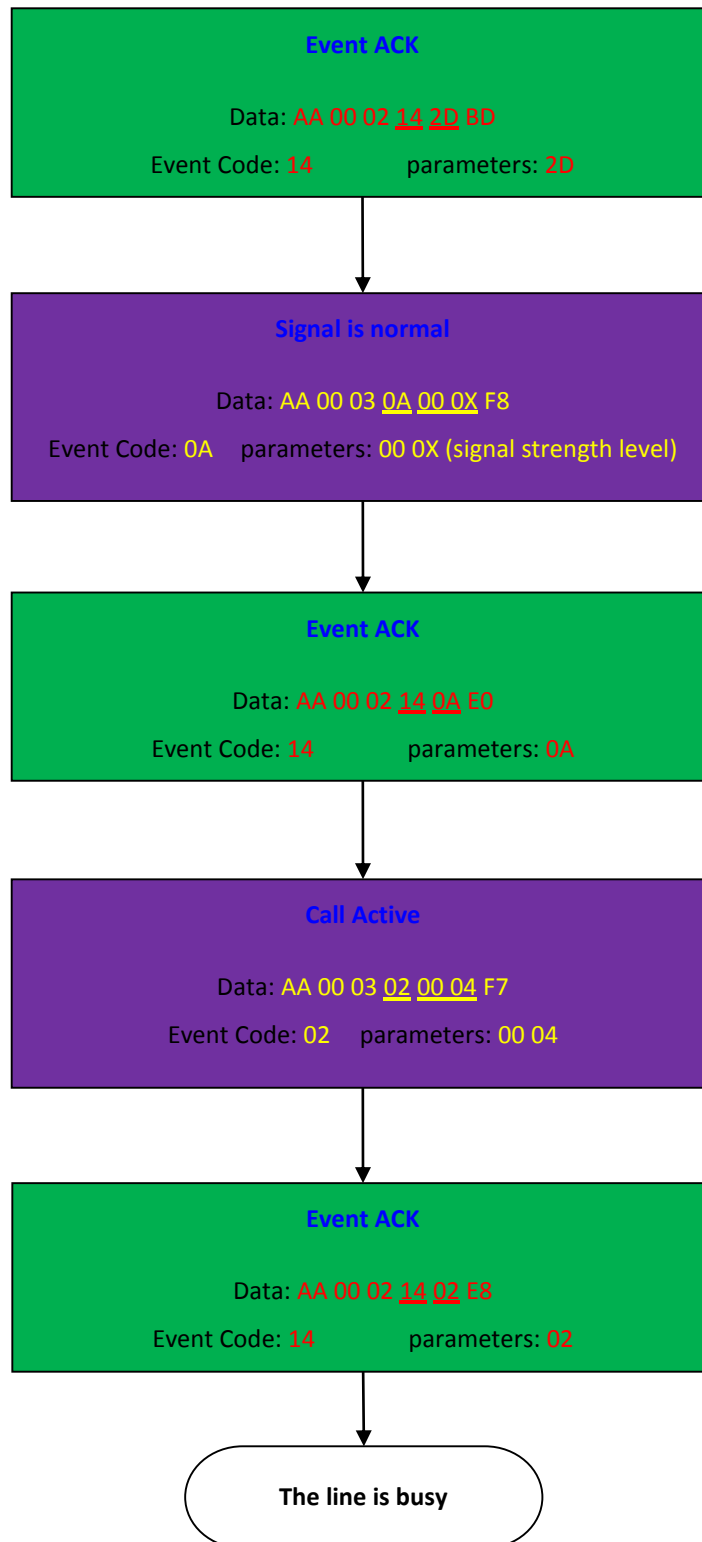


■ : SPK Module → MCU (No Command Action)

■ : MCU → SPK Module

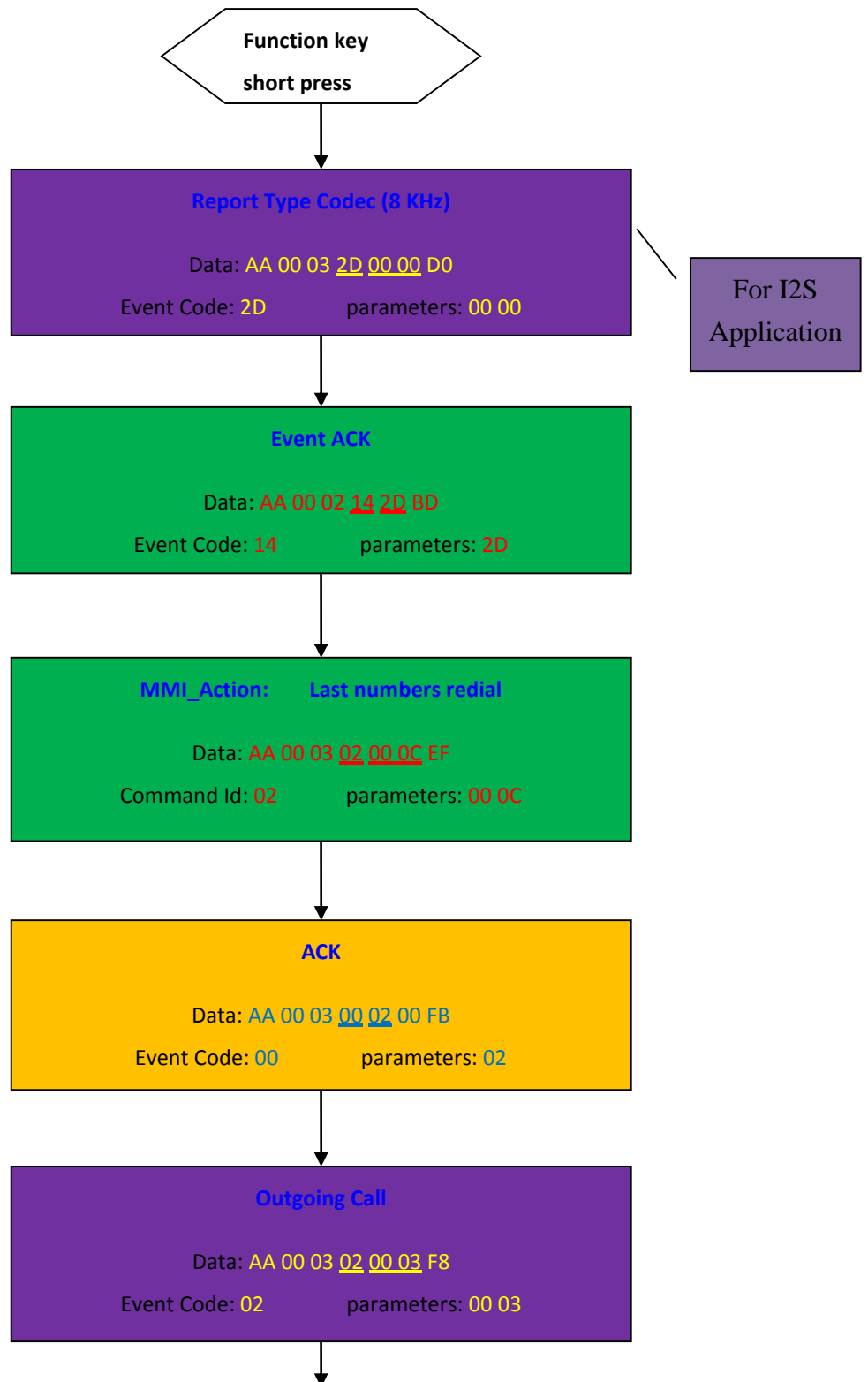
**Dial (Hands-free operate, HF connected)**

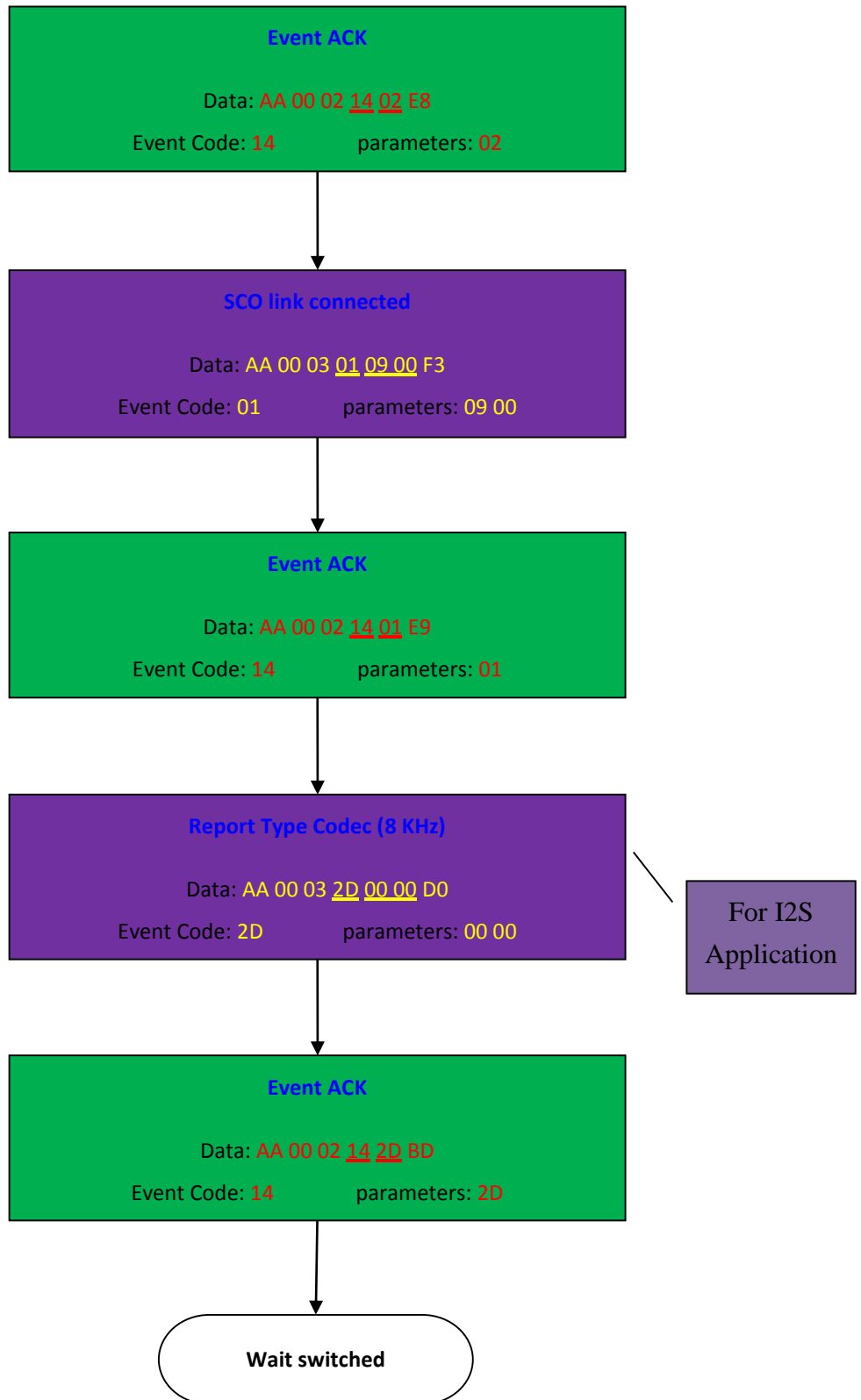




- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

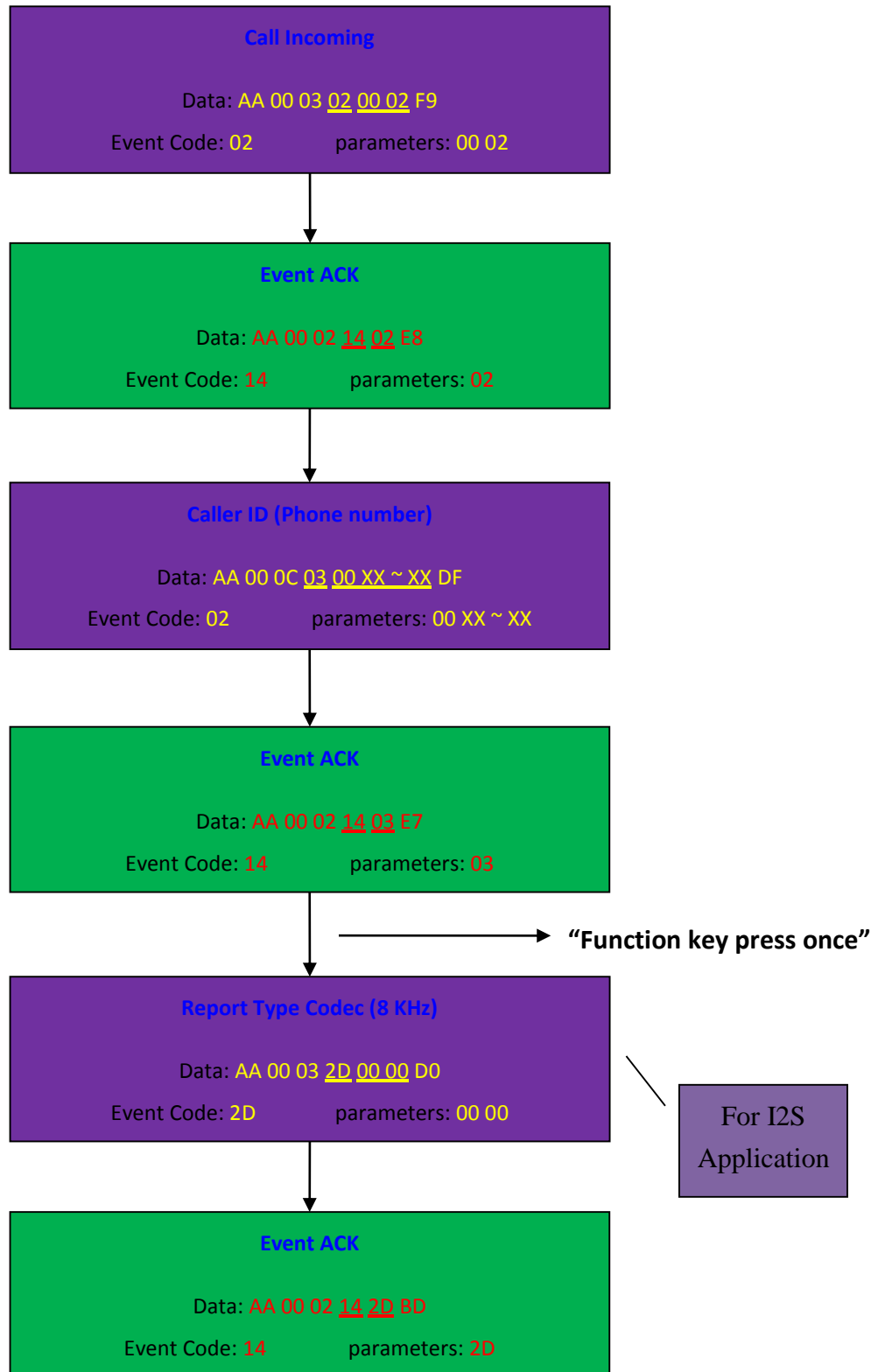
**Re-Dial (@BT Mode, HF connected)**

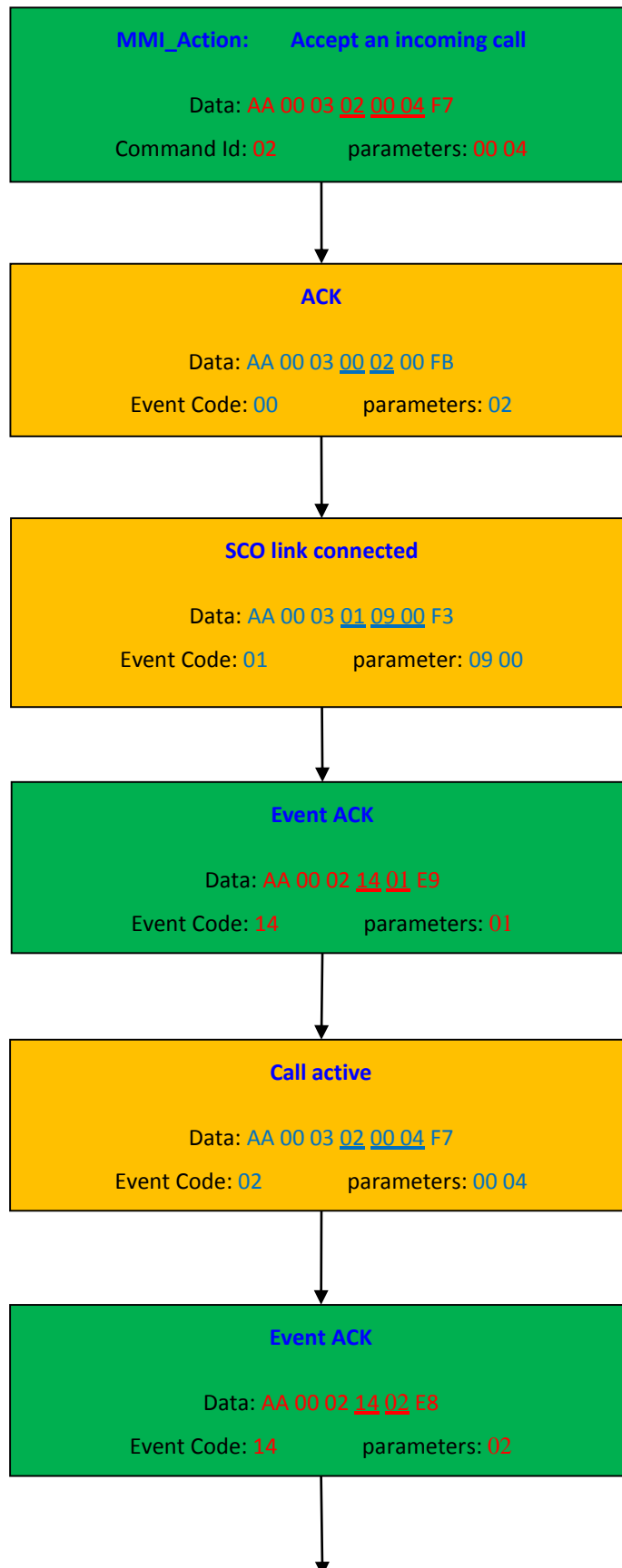


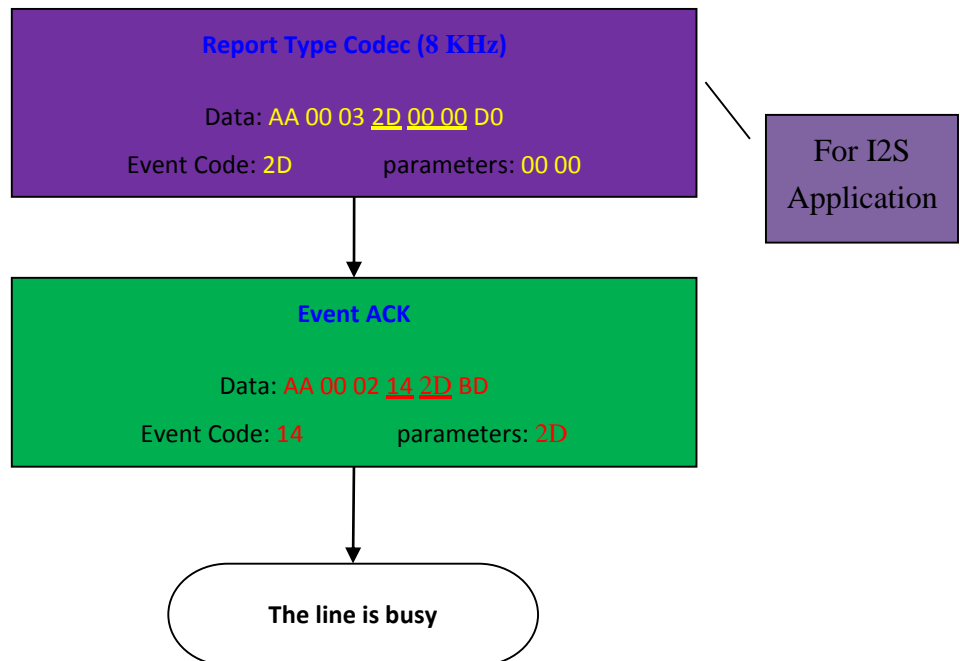


- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

**Incoming (HF connected)**

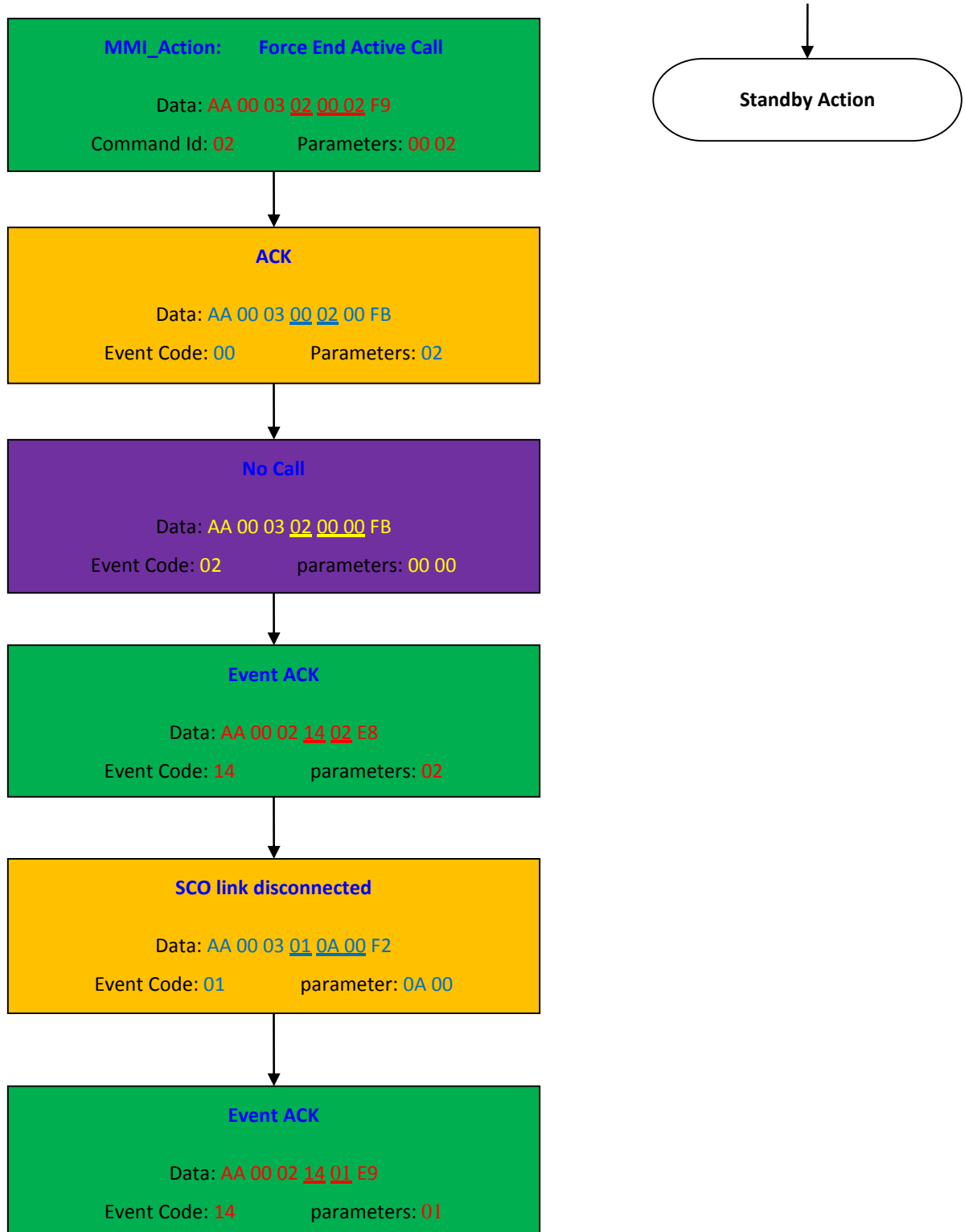






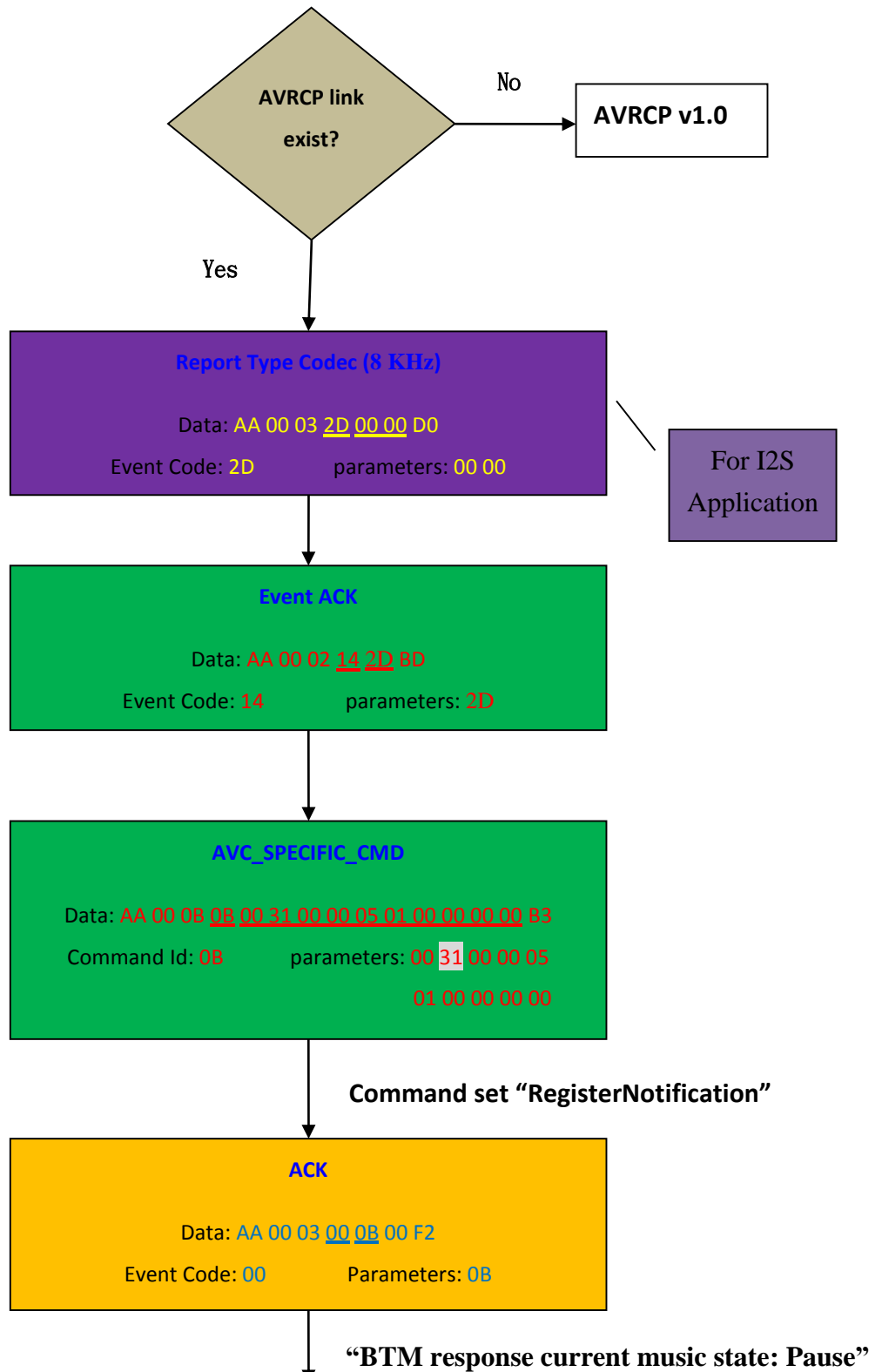
- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

**Ring off (HF connected)**

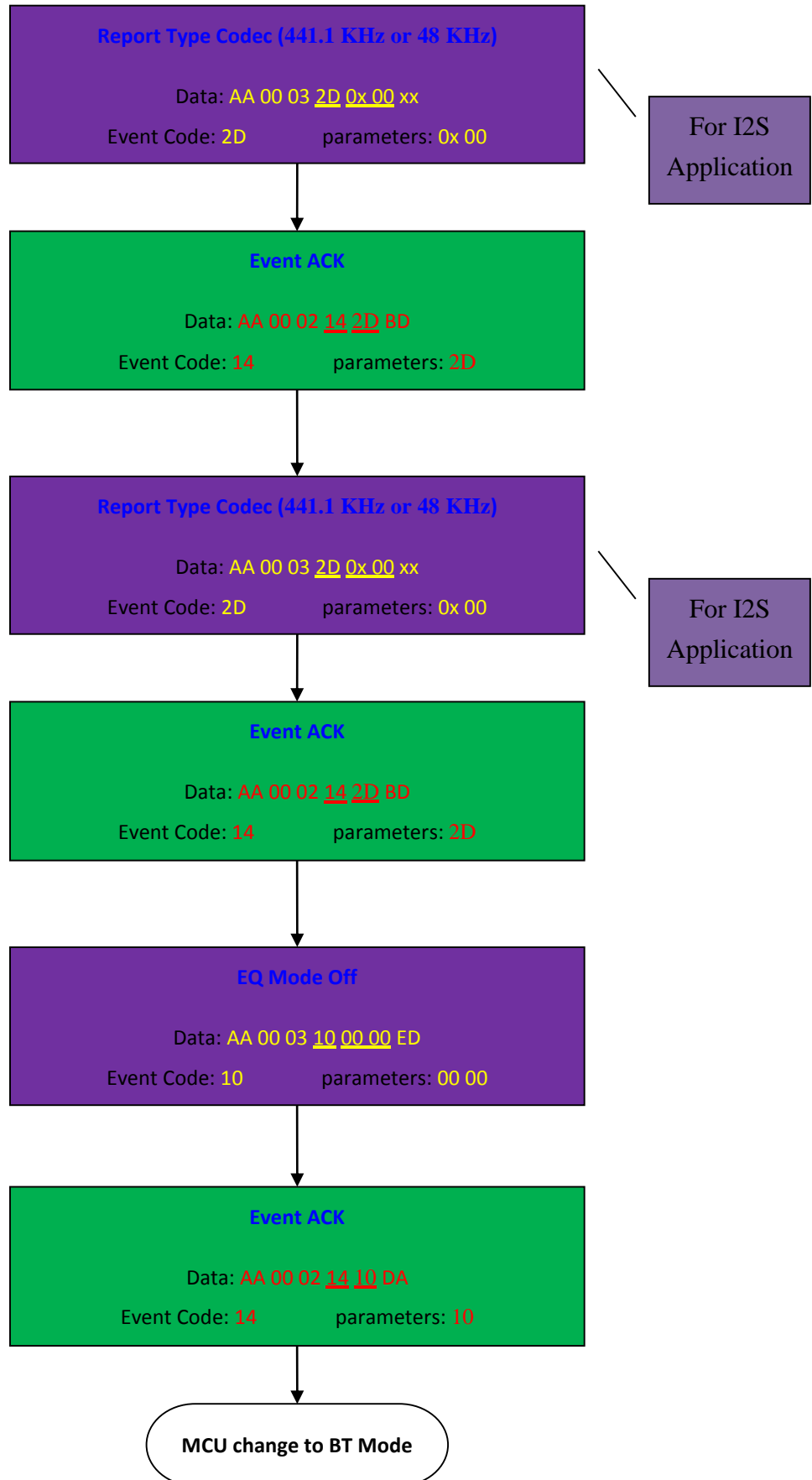


- : MCU → SPK Module
- : SPK Module → MCU (Command Accept)
- : SPK Module → MCU (No Command Action)

**AVRCP actions (support AVRCP v1.3~ ver.)**







#### 5.4.2 RegisterNotification (PDU ID: 0x31)

Description:

This primitive registers with the TG to receive notifications asynchronously based on specific events occurring. The initial response to this Notify command shall be an INTERIM response with current status, or a REJECTED/NOT IMPLEMENTED response. This has to take place within  $T_{MTP}$  time from receiving the command. The following response shall be a CHANGED response with the updated status, or a

16 April 2007

REJECT response. This is as per 1394 AV/C protocol specification. A registered notification gets changed on receiving CHANGED event notification. For a new notification additional NOTIFY command is expected to be sent. Only one EventID shall be used per notification registration.

Refer to [Figure 25.2](#) in Appendix J: List of Example MSC of different Metadata Transfer Commands.

Command Format (RegisterNotification):

Parameters	Size(byte)	Description	Allowed Values
EventID	1	Event for which the CT requires notifications	see Error! Reference source not found.
Playback interval	4	Specifies the time interval (in seconds) at which the change in playback position will be notified. If the song is being forwarded / rewound, a notification will be received whenever the playback position will change by this value. (Applicable only for EventID EVENT_PLAYBACK_POS_CHANGED. For other events , value of this parameter is ignored)	$0 < \text{Playback interval}$

Table 5.27: RegisterNotification command

### Allowed Values for EventID

EventID	Description
EVENT_PLAYBACK_STATUS_CHANGED (0x01)	Event for change in playback status
EVENT_TRACK_CHANGED (0x02)	Event for change in track
EVENT_TRACK_REACHED_END (0x03)	Event for reach to end of the current track
EVENT_TRACK_REACHED_START (0x04)	Event for reach to start of the current track.
EVENT_PLAYBACK_POS_CHANGED (0x05)	Event for change in playback position
EVENT_BATT_STATUS_CHANGED (0x06)	Event for change in battery status
EVENT_SYSTEM_STATUS_CHANGED (0x07)	Event for change in system status
EVENT_PLAYER_APPLICATION_SETTING_CHANGED (0x08)	Event for change in player application setting

Table 5.28: Allowed Values for EventID

### Response Formats (RegisterNotification)

#### Response Data format for EVENT\_PLAYBACK\_STATUS\_CHANGED

Parameters	Size(byte)	Description	Allowed Values
EventID	1	Specific EventID	EVENT_PLAYBACK_STATUS_CHANGED (0x01)
PlayStatus	1	Indicates the current status of playback	0x00: STOPPED 0x01: PLAYING 0x02: PAUSED 0x03: FWD_SEEK 0x04: REV_SEEK 0xFF: ERROR

Table 5.29: Response EVENT\_PLAYBACK\_STATUS\_CHANGED